PRIORITY-BASED SCHEDULING POLICIES FOR BLUETOOTH

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by

D. Raveendra Babu

to the

Department of Electrical Engineering
Indian Institute of Technology, Kanpur

May, 2001

Certificate

This is to certify that the work contained in the thesis entitled "*PRIORITY-BASED SCHEDULING POLICIES FOR BLUETOOTH*", by *D. Raveendra Babu*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

May, 2001

(Vishwanath Sinha)
Professor,
Department of Electrical Engineering,
Indian Institute of Technology,
Kanpur.

Dedicated
to

My Parents, Sister and Brother.

Acknowledgements

I take this opportunity to express my sincere gratitude towards Prof. V. Sinha, my thesis supervisor for his invaluable guidance. He has always been very patient and encouraging regarding the doubts and discussions during the thesis work. I would like to thank Dr. R.k. Bansal, Prof. S.K. Bose, Dr. A.k. Chaturvedi, Dr. Sumana Gupta, Dr. Y.N. Singh, Dr. Umesh for teaching me various courses in field of communication.

I am thankful to my batch mates Amit, Anil Pandey, Bhoomika, Raju, Sasi, Sood, Srilakshmi, Suresh and Venu for their suggestions and maintaining good work environment in ERNET lab. Also, I would like to thank all my friends of MTech'99 and juniors (MTech'00) for making my stay at IITK a memorable one. I would like to mention the support of Vineet, Vivek and other research staff of network floor.

Words are not enough to explain my feelings towards my parents, sister and brother for their love and inspiration in all ventures of my life.

Abstract

Bluetooth is a wireless adhoc network concept primarily intended to eliminate the cables between computers, cellphones, PDAs etc. The Bluetooth radio nodes form adhoc networks called piconets. A Bluetooth unit can participate in more than one piconet at any time but it can be a master in only one piconet. A unit that participates in multiple piconets can serve as a bridge thus allowing the piconets to form a larger network. A set of piconets that are all interconnected by such bridging units is referred to as a scatternet network.

In any communication system, it is important to support various traffic with QoS (Quality-of-Service) gaurantees. Broadly, diverse traffic may be categorized as real-time traffic and non-real-time traffic. Bluetooth supports both real-time and non-real-time communication services. A non-real-time communication service in Bluetooth is considered. Depending on its distinct characteristics and QoS requirements, the non-real-time traffic can be divided into two classes: a) class1: delay-tolerant traffic like paging and email; and b) class2: delay-sensitive traffic like FTP and remote log-in. The main distinguishing factor between these classes is how delay sensitive the traffic is. The purpose of present work is to support different classes of service (class1 and class2) in Bluetooth through priority and scheduling mechanisms. In our proposed methods, class2 traffic is given priority over class1. The two important parameters that have been considered are minimizing end-to-end packet delivery delay and providing consistent data throughput and capacity. We examine in detail the performance of non-real-time communication service in Bluetooth.

Contents

List of Tables

List of Figures

List of Acronyms

1.	PDA	Personal Digital Assistant
2.	ISM	Industrial Scientific Medical
3.	IrDA	Infrared Data Association
4.	OBEX	Object Exchange Protocol
5.	DS-WLAN	Direct Sequence - Wireless Local Area Network
6.	TDD	Time Division Duplexing
7.	QoS	Quality of Service
8.	FTP	File Transfer Protocol
9.	RF	Radio Frequency
10.	FH-CDMA	Frequency Hopping - Code Division Multiple Access
11.	GFSK	Gaussian Frequency Shift Keying
12.	ID	Identification
13.	FHS	Frequency Hopping Synchronization
14.	SCO	Synchronous Connection Oriented
15.	ACL	Asynchronous Connection Less
16.	MAC	Medium Access Control

17. RR	Round Robin
18. PQ	Priority Queueing
19. PP	Priority Polling
20. FIFO	First In First Out
21. FCFS	First Come First Serve
22. LCFS	Last Come First Serve

Chapter 1

Introduction

Bluetooth [1] is a short range radio link technology primarily developed to eliminate the wireline connection between electronic devices, such as computers, mobile phones, PDAs and printers. The technology is an open specification for wireless communication of data and voice. The Bluetooth system operates in the unlicensed 2.45GHz ISM (Industrial-Sceintific-Medical) band.

Apart from cable replacement, Bluetooth enables Bluetooth-equipped hosts to connect and communicate amongst themselves in an adhoc fashion. Bluetooth adopts a master-slave configuration to form restricted types of adhoc networks called piconets. Typically many independent networks overlap in the same area which will be indicated as scatter adhoc environment [2]. Routing protocols that have been devised for routing in conventional adhoc networks, [3, 4, 5], may not be efficient solutions for routing in the Bluetooth scatternets. Because scatternets differ from classical adhoc networks in terms of the applications, traffic characteristics, mobility patterns and scaling requirements. There has been a recently proposed algorithm [6] for routing in Bluetooth system considering the similar parameters of interest. The objective was to design a simple and bandwidth efficient protocol.

Since Bluetooth operates in the unlicensed ISM band along with other systems, a major concern is the amount of interference that these systems introduce to each other, thereby influencing their performance. The effect of interference caused by a Bluetooth system on IEEE 802.11 is studied in [7]. While the opposite case,

Bluetooth voice and data performance in 802.11 DS WLAN, is studied in [8]. The simulation results presented in [8] showed that, in an office environment, a Bluetooth voice link was disturbed in less than 1% of time when the distance between devices was less than 2m but increased to 8% for distances between 2m and 10m.

Another short range wireless technology which was intended for cable replacement is Infrared Data Association (IrDA). The higher layer protocol, Object Exchange Protocol (OBEX), developed and used by IrDA may also be used by Bluetooth usage models since it is a part of Bluetooth protocol stack. Hence, it is possible for an application to run over both IrDA and Bluetooth. The better choice between the two technologies depends on a particular application. A comparison is presented in [9].

Bluetooth uses centralized TDD scheduling (i.e., at the master) as the MAC protocol for scheduling access to the wireless medium. Master present in the piconet schedules the traffic in both the uplink and downlink. Efficient scheduling algorithms need to be designed that take into account the slave characteristics. Two new scheduling policies for Bluetooth were proposed in [10] to increase the achievable channel utilization (throughput) and fairness. The policies were based on the size of the Head-of-Line packet at the master and slave queues.

In any communication system, it is important to support various traffic with QoS (Quality-of-Service) gaurantees. To satisfy the QoS needs of differnet connections, nodes present in the network need to have priority and scheduling mechanisms. The priority feature typically refers to the capability of providing different delay treatment, e.g., higher priority packets are always served before the lower priority ones. Whereas scheduling feature refers to how the network elements handle the overflow of arriving traffic using a particular queueing algorithm and then determining some method with which queued packets are selected for transmission on the link.

1.1 Objective of the Thesis:

Broadly, diverse traffic may be categorized into two types: 1) real-time traffic such as voice that requires bounded delays; 2) non-real-time traffic like the conventional data

services that requires no bounded delays [11]. Bluetooth supports both real-time (voice) and non-real-time (data) communication services. A non-real-time communication serice in Bluetooth is considered. Depending on its distinct characteristics and QoS requirements, the non-real-time traffic can be divided into two classes: a) class1: delay-tolerant like paging and email; and b) class2: delay-sensitive traffic like FTP and remote log-in [11]. The motivation behind our work is to support different classes of service (class1 and class2) in Bluetooth through priority and scheduling mechanisms. The objective is to minimize the end-to-end packet delivery delay and providing consistent data throughput and capacity for a delay-sensitive traffic like class2 traffic. In our proposed methods class2 traffic is given priority over class1 traffic. We examine in detail the performance of non-real-time communication service in Bluetooth.

This thesis is organized as follows. We briefly describe Bluetooth system characteristics in chapter 2. Deriving the mean delays by analitical modeling is presented in Chapter 3. Simulation results are given in Chapter 4. Chapter 5 presents the conclusion and future work.

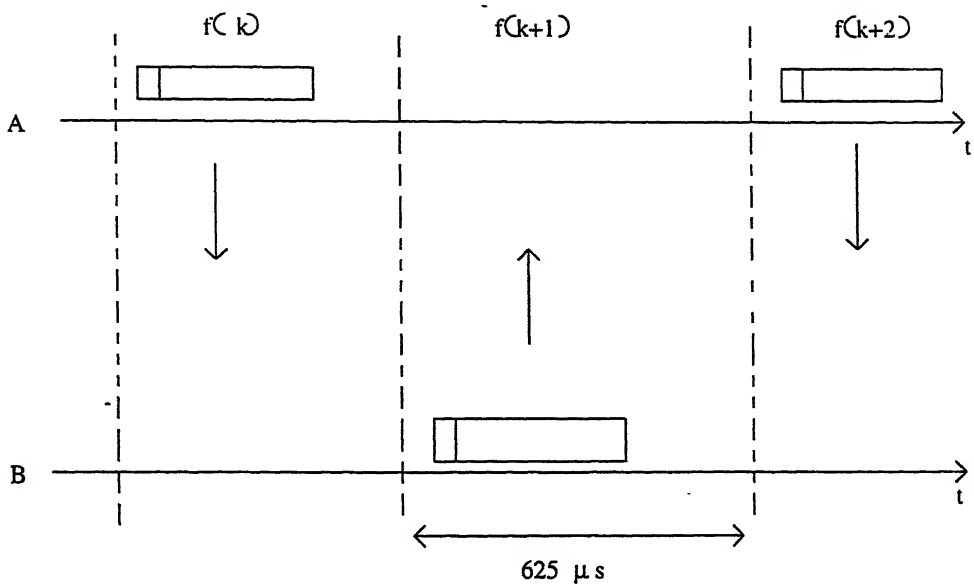
Chapter 2

Bluetooth System Characteristics

Bluetooth operates in the unlicensed ISM band at 2.45 GHz. This operational frequency band is divided into 79 (in US and most of Europe) or 23 (in Japan, Spain and France) RF channels spaced 1 MHz apart. The technology uses FH-CDMA as a multiple access scheme.The frequency hopping scheme is adopted to make Bluetooth more insensitive to interference from other devices operating in the same band. Modulation scheme employed in Bluetooth is Gaussian Frequency Shift Keying (GFSK).

2.1 Bluetooth channel

In Bluetooth, the 79 or 23 RF channels, each of 1 MHZ bandwidth, are accessed in a pseudo-random hopping manner. So the Bluetooth channel is represented by a pseudo-random hopping sequence hopping through the 79 or 23 RF channels. In time domain, the channel is divided into time slots of 625μs in length, i.e., 1600 slots per second. The frequency changes each time a new time slot or hop begins. On the channel, information is exchanged through packets. Each packet is transmitted on different hop frequency. The process of communication between two Bluetooth units (A and B) is shown in Fig. 2.1. A packet nominally covers one slot. But it can be extended to cover three or five slots in which case, the used frequency stays the same for the duration of the packet rather than changing on per slot basis, see Fig.



$625\mu s$

Figure 2.1: An illustration of the FH/TDD channel applied in Bluetooth

2.2. For full duplex transmisson, a Time-Division Duplex (TDD) scheme is used.

2.2 Piconet

The Bluetooth system provides a point-to-point connection (only two Bluetooth units involved), or a point-to-multipoint connection. In the point-to-multipoint connection, the channel is shared among several Bluetooth units. Two or more units sharing the same channel form a piconet. One Bluetooth unit acts as the master of the piconet, whereas the other units act as slaves. Up to seven slaves can be simultaneously active in the piconet. Slaves are allowed to exchange packets only with their master.

2.2.1 Piconet Formation

Each Bluetooth host is assigned a unique 48-bit Bluetooth device address (BD_ADDR) derived from the IEEE 802 standard. Also, each host has a free-running clock (i.e a

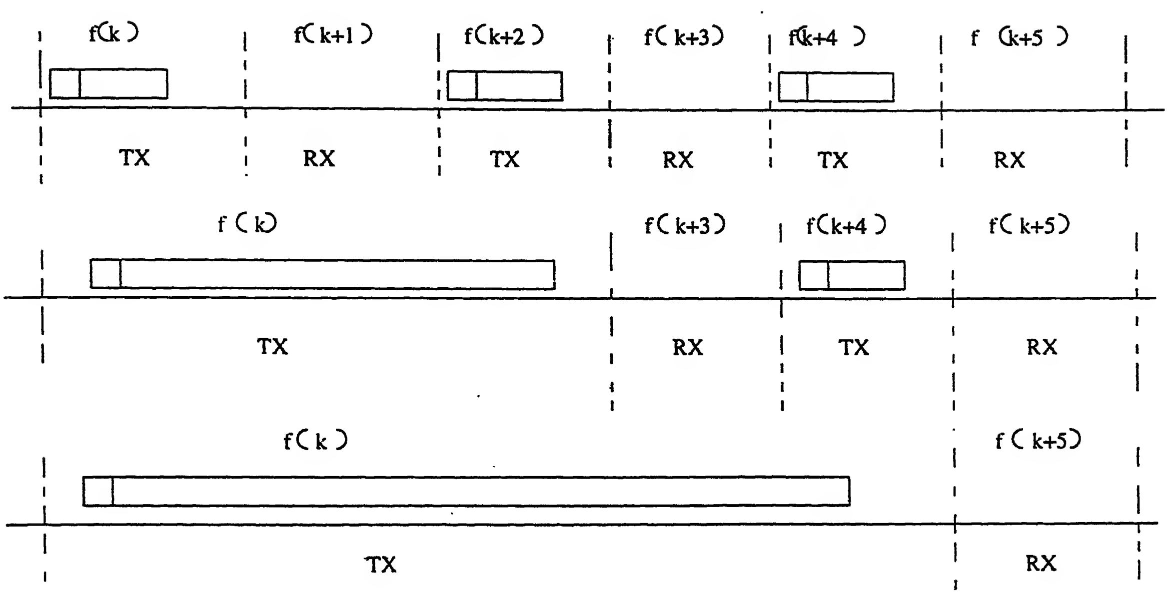


Figure 2.2: The frequency and timing characteristics of single-slot, three-slot, and five-slot packets

clock that is not related to any external time source) that ticks every 312.5μs. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that wraps around at $2^{28} - 1$.

The Bluetooth device address defines the piconet channel (channel hopping sequence). So each Bluetooth unit has its own pseudo-random hopping sequence which is determined by its unique identity. The sequence is cyclic, but with a peroid of more than 23 hours. These long sequences prevent periodic collisions between uncoordinated Bluetooth connections. The phase in the sequence is determined by the unit's system clock.

To form a piconet, at first all the units participating in a piconet need to select a particular channel and subsequently join in it. So the piconet channel is defined by the identity (providing the hop sequence) and system clock (providing the hop phase) of a master unit. All the slaves paricipating in the piconet are time and hop-synchronized to the channel. This process of synchronization is explained in the following section. This implies that the channel in the piconet is characterized entirely by the master of the piconet.

2.3 Connection Establishment

The Bluetooth specification defines two major states (modes of operation) for a unit:

- Standby
- Connection.

Each unit that communicates with other Bluetooth units is a member of piconet and is in the Connection state. Establishment of a piconet is initiated by any unit that wants to communicate with other ones and it becomes the master of piconet. Any communication between hosts using Bluetooth is preceded by a connection establishment procedure.

In the connection establishment procedure, hosts are hopping independently and they have to meet on a same frequency to be able to transmit and receive data necessary for synchronization and forming a piconet (Bluetooth device address and native clock values). The connection establishment procedure consists of two steps: Inquiry and Page. There are seven substates that are used in these procedures. Fig 2.3 shows all the the states with the transitions between them as given in [1].

In the following section we describe the details of Bluetooth connection establishment procedure through the description of states and the transitions shown in Fig 2.3. Also in the following section, the device initiating the connection, and which becomes the master is referred as P, The second device which becomes slave is referred as Q.

2.4 Overview of States

2.4.1 Standby

Standby is the default state of a Bluetooth unit . It is a low power mode in which only the native clock is running. A unit might leave Standby to go to inquiry, page, inquiry scan or page scan.

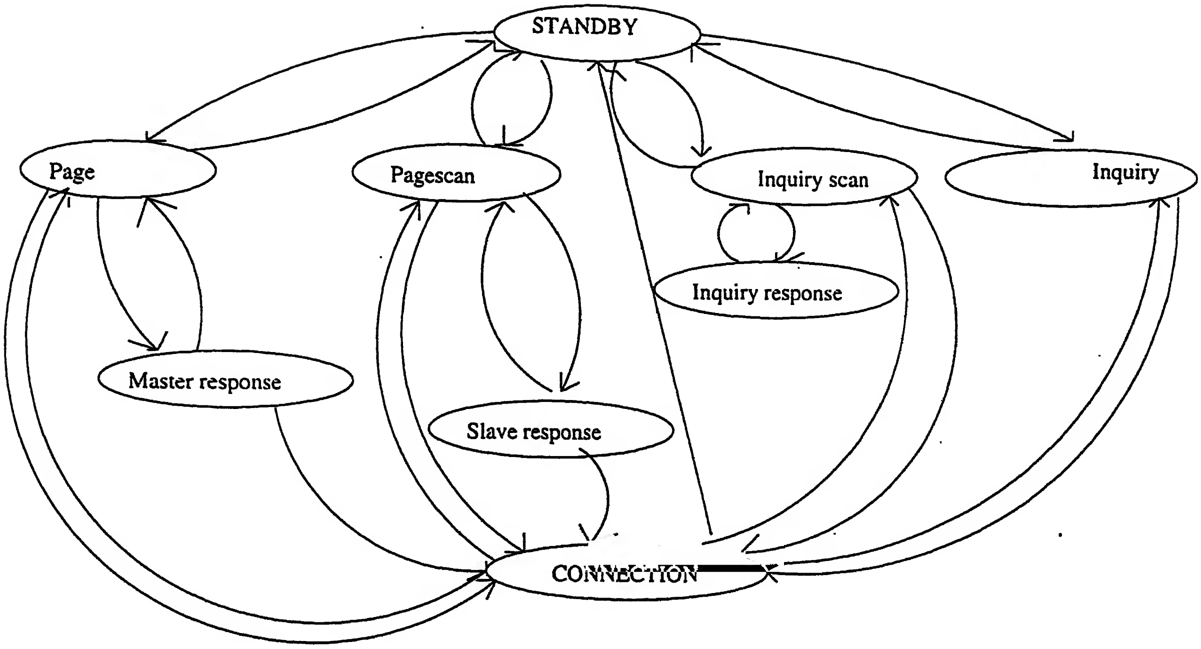


Figure 2.3: Bluetooth unit states

2.4.2 Inquiry

The Inquiry procedure enables a unit (P) to discover which units are in range, and what are their synchronization information. After a successful inquiry, a master will have Bluetooth device address and the value of their native clocks. The procedure is as follows: The node P sequentially transmits two inquiry messages (broadcast ID packets) on two differnet frequencies in each TX slot. Now the hopping rate is increased to 3200 hops/sec. In each Rx slot, the unit listens for a response sequentially on two corresponding frequencies. Unit P continuously alternates between TX and Rx slots. For inquiry messages, an *inquiry sequence* containing 32 unique, dedicated, prespecified frequencies is used. The sequence is split into two trains A and B each with 16 freqencies. Each train must be repeated 256 times to collect all responses in an error free environment. For inquiry message responses sent(by Q) in the RX slots, the *inquiry response sequence* is used, covering 32 frequencies that are in one-to-one correspondence to the frequencies in the inquiry sequence. A response is a Frequency Hopping Synchronization (FHS) packet which contains the

Bluetooth device address and the native clock of the responding unit.

2.4.3 Inquiry Scan

Inquiry scan is the substate that enables a unit (Q) to be discovered by other units (P). For this a device will periodically enter the inquiry scan substate and listen for inquiry messages at a single frquency chosen from 16 frquencies in the inquiry sequence.

2.4.4 Inquiry Response

When a unit (Q) received an inquiry message in inquiry scan state, a response packet called Frequncy Hopping synchronization (FHS) packet must be sent. The FHS packet contains the device address, its clock and information about when the device enters page scan states. This packet is sent on frequency from the inquiry response sequence corresponding to the frequency on which an inquiry message is received. But the units are not allowed to respond immediately after receiving inquiry message to avoid collisions of packets sent simulataneously by a other units using the same frequency.

2.4.5 Page

The page state is used by a future master (P) to set up a connection with a particular unit (Q). After the inquiry has been successfully carried out, a unit (P) will have the device address of the unit (Q) to which a connection has to be made. Now a device will start paging procedure if a connection is desired. Paging requires only the address of the device to be paged but the clock information from the FHS response packet may be used to speed up the procedure. The device starting the paging procedure is called the master, and it will be the master of the piconet. The paging procedure is the same as the inquiry procudure; however instead of sending broadcast ID packets, a master sends ID packets to the slave to which it wants to connect. This ID packet uniquely identifies future slave (Q) since its content is derived from the slave's address.

A paging unit (P) uses a *Page hopping sequence* containing 32 frequencies assigned to paged unit (Q). Unlike the case of inquiry where there is a inquiry sequence, a page sequence is different for each unit and is determined by the address (BD_ADDR) of that unit. So a master will use address and clock of unit to which it wants to connect. There is a corresponding *page response sequence* for page response.

2.4.6 Page Scan

In the page scan substate, a unit (Q) listens for the ID packets which contains its own address and unicast to it by the master (P). Similar to inquiry scan, the terminal (Q) listens on one frequency chosen from its page hopping sequence and based on its native clock at the start of the page scan. When an ID packet is received, the host (Q) enters the slave response state.

2.4.7 Master Response

The master response state is entered by a unit (P) after it has received a response form slave (Q) after page scan mode. Master (P) will send a FHS packet to the prevoiusly paged unit (Q) and waits for an acknowledgement. After an acknowledgement has received, the master enters the connection state. From now onwards, channel hopping sequence will be used.

2.4.8 Slave Response

A unit (Q) enters the slave response state form page scan when a page message (unicast ID) has been received. The unit (Q) sends a response using same ID packet and waits for the response message (FHS) packet from the master (P). After getting the FHS packet form master (P), it (Q) sends the acknowledgement and switches to the connection state. From now onwards, it (Q) also uses channel hopping sequence. The flow of messages during the connection establishment procedure is shown in Figure 2.4.

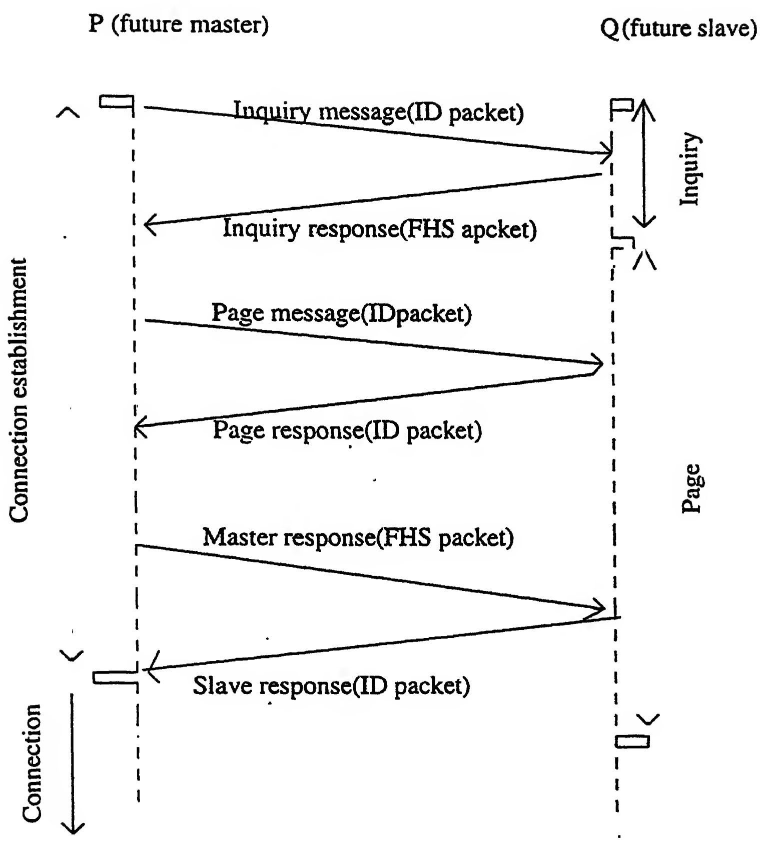


Figure 2.4: Flow of messages in connection estabblishment procedure

2.4.9 Connection

A unit in connection state is a part of piconet. To transmit or receive data, a channel hopping sequence which is determined by the master's address and clock is used.

The Bluetooth units can be in several modes of operation during the connection state: *activemode, sniffmode, holdmode, park mode.* These modes are described in the following.

- Active mode

  In the active mode, the Bluetooth unit actively participates in packet exchange in a piconet. The master schedules the transmission based on traffic demands to and from the slaves. The Master uses polling [12] to schedule the transmission from slaves. slave can send packets in only the slots assigned to it by the master. There can be upto seven active slaves in a piconet.

- Sniff mode

  In the sniff mode, the duty cycle of the slave can be reduced. When a slave is in sniff mode, its master can only transmit to that slave in specified time slots that are regularly spaced.

- Hold mode

  In hold mode, a slave doesn't support packet exchange, but it stays as a member of a piconet. Prior to entering the hold mode, master and slave agree on the time duration the slave stays in the hold mode. With the hold mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet.

- Park mode

  Park mode is low power mode in which a slave doesn't participate in the packet exchange, but remain synchronized to the channel. The parked slave wakes up at regular intervals to listen to the channel.

2.5 Medium Access Control

Medium access control refers to allocating the multiaccess channel so that each node can successfully transmit its packets without undue interference from others. Bluetooth uses centralized TDD scheduling (i.e., at the master) as the MAC protocol for scheduling access to the wireless medium.

In Bluetooth, master implements centralized control; only communication between the master and one or more slaves is possible. The TDD slot structure gives a complete contention free channel. On a Bluetooth channel, time slots are devided into two groups: master-to-slave and slave-to-master slots. There is a strict alternation of slots. The master can only send packets to a slave in the even slots while the slave can send packets to the master in odd slot [1]. In the master tranmission, the master uses the slave address of the unit to which the packet is intended. But in the slave-to-master slots, in order to prevent collisions on the channel due to multiple slave transmissions master applies POLLing scheme.

- Polling: For each slave-to-master slot, the master decides which slave is allowed to transmit. A slave may transmit the packet in the slave-to-master slot only if it is addressed by the master in the preceding master-to-slave slot. In polling a particular slave, if the master has any information intended for that slave it sends the packet in master-to-slave slot and polling will be done implicitly. Whereas if the master doesn't have any information, it polls the slave explicitely by sending simple poll packet.

The Bluetooth protocol uses a combination of cicuit switching and packet switching. A circuit switched connection is created by a link called Synchronous Connection Oriented (SCO) link, whereas packet switched connection by Asynchronous Connection Less (ACL) link. Both link types use a packet based transport mechanism. The SCO link is used for time-bounded data such as voice whereas ACL link is used by conventional data services. SCO link is a point-to-point link between the master and single slave and it is implemented by reserving duplex time slots at regular intervals. The ACL link is point-to-multipoint link between the master

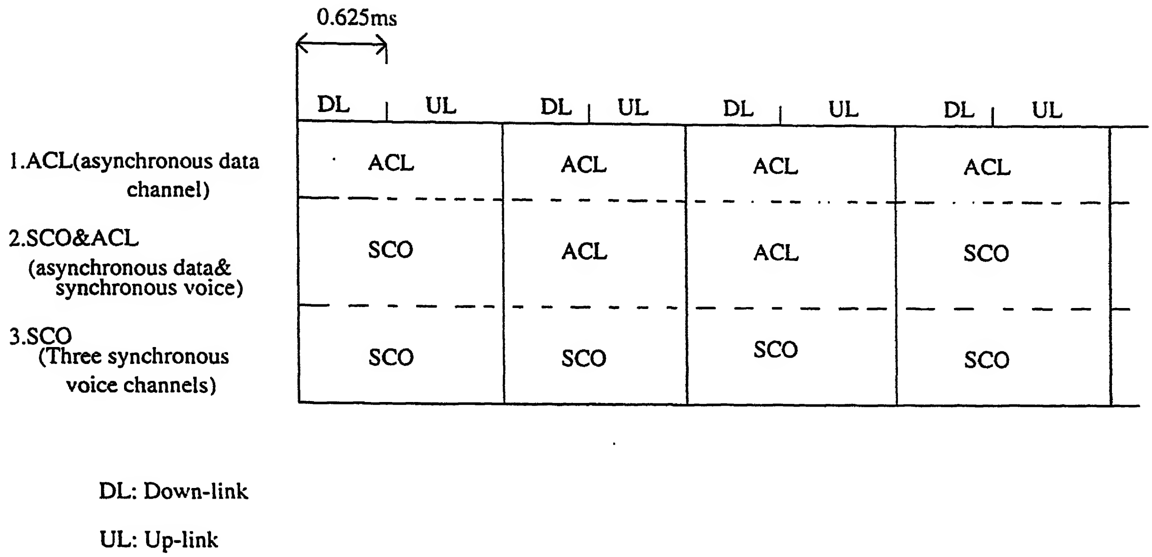


Figure 2.5: Bluetooth radio channel time slots showing SCO/ACL link combinations

and all the slaves present in the piconet. In the slots not reserved for SCO link the master establishes ACL link on a per slot basis to any slave.

Bluetooth can support an asynchronous data channel (ACL link), up to three synchronous voice channels (SCO links) or a channel which simulataneously supports asynchronous data and synchronous voice. Fig 2.5 illustrates how the slotted channel can be used for different combinations of SCO and ACL links. Each voice channel supports a 64 kb/s synchronous link. The asynchronous channel can support a maximal 721kb/s asymmetric link (and 57.6 kb/s in other direction), or a 432.6 kb/s symmetric link.

2.6 Interpiconet Communication

Typically many independent piconets may overlap in the same area. These multiple piconets with overlaping area of coverage can co-exist since their frequency hopping patterns are mutually orthogonal. A Bluetooth unit can participate in more than one piconet on a time sharing basis, i.e., at any instant the node can be active in only one piconet. Such a unit can receive packets from one piconet and relay them

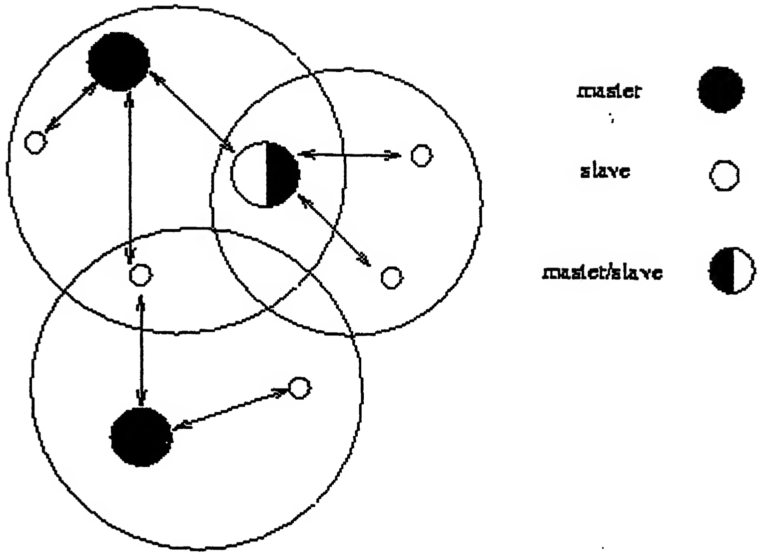


Figure 2.6: Bluetooth scatternet

to the other piconets it is connected to. The hop selection mechanism has been designed to allow for interpiconet communication. When a unit jumps from one piconet to another, it changes the identity and clock input (channel parameters) to the hop selection mechanism. So instantaneously a new hop for the new piconet is selected.

A unit that participates in multiple piconets can serve as a bridge thus allowing the piconets to form a larger networks. These bridging units, when they are jumping to the new piconet will request to enter the *hold or park* mode in the current piconet. A set of piconets that are all interconnected by such bridging units is referred to as a *scatternet*. A scatternet is shown in Fig 2.6.

Chapter 3

Analytical Modeling

Bluetooth uses master driven Time Division Duplexing (TDD) system at the Medium Access control (MAC) layer to support full duplex transmission. The master polls the slaves (either implicitly or explicitly) in the even slots while the slave sends the packets in odd slots. This implies that the scheduling occurs in pairs of slots and also the task of scheduling is vested at the master. This kind of master driven scheduling at the MAC level affects the throughput and queueing delay. When multiple data transfers with various QoS requirements share the wireless link, MAC scheduling algorithms are needed to achieve fair sharing of bandwidth with QoS gaurantees.

One of the conventional scheduling algorithms, Round-Robin (RR) scheduling can be used to schedule the data in a piconet, because it is simple, fair, and widely used scheduling algorithm. Using round-robin scheduling, it is possible to provide each slave a fair access to the channel. In round-robin scheduling amongst slaves, each master-slave connection is alloted a pair of slots.

In the following section, we carry out the analysis for delay of a packet in both uplink and downlink using round-robin scheduling algorithm. We consider a piconet consisting of one master and seven slaves for this analysis. There exists a separate queue at the master for each slave. Here, the master could be the Fixed Access point or the Base station. The analysis is done using the "analysis for reservations and polling systems" given in [12]. We have modified the analysis to specifically suit the Bluetooth environment. We examine in detail the performance of non-real-time

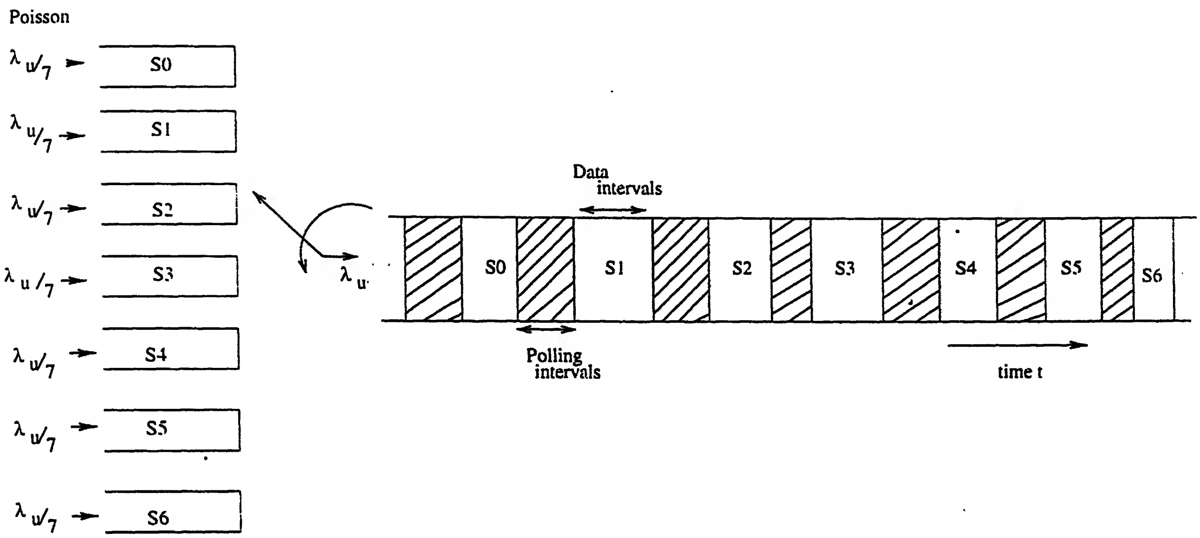


Figure 3.1: Illustration of channel in the uplink with Round-Robin scheduling

communication service.

3.1 Delay analysis using Round-Robin scheduling:

3.1.1 Uplink:

In the uplink, the communication resource of the channel can be divided over time into a portion used for packet transmission from the slave queue and another portion used for polling messages coming from the master that coordinates the packet transmissions. In other words, the time axis is divided into *data intervals*, where actual data is transmitted, and *polling intervals*, used for scheduling future data. The polling intervals are corresponding to packet transmission intervals from master to slaves, whereas the data intervals are packet transmission intervals from slaves.

We will consider m (seven) traffic streams (packets arriving at slave queues). Each data interval contains single packet from a single slave queue. Reservation for this packet is made in the immediately preceding polling interval. All the slaves are taken up in cyclic order (Round-Robin), see Fig. 3.1.

We assume that the arrival process of the packets at each slave queue is Poisson with an arrival rate of λ_u/m, and that the first and second moments of the service

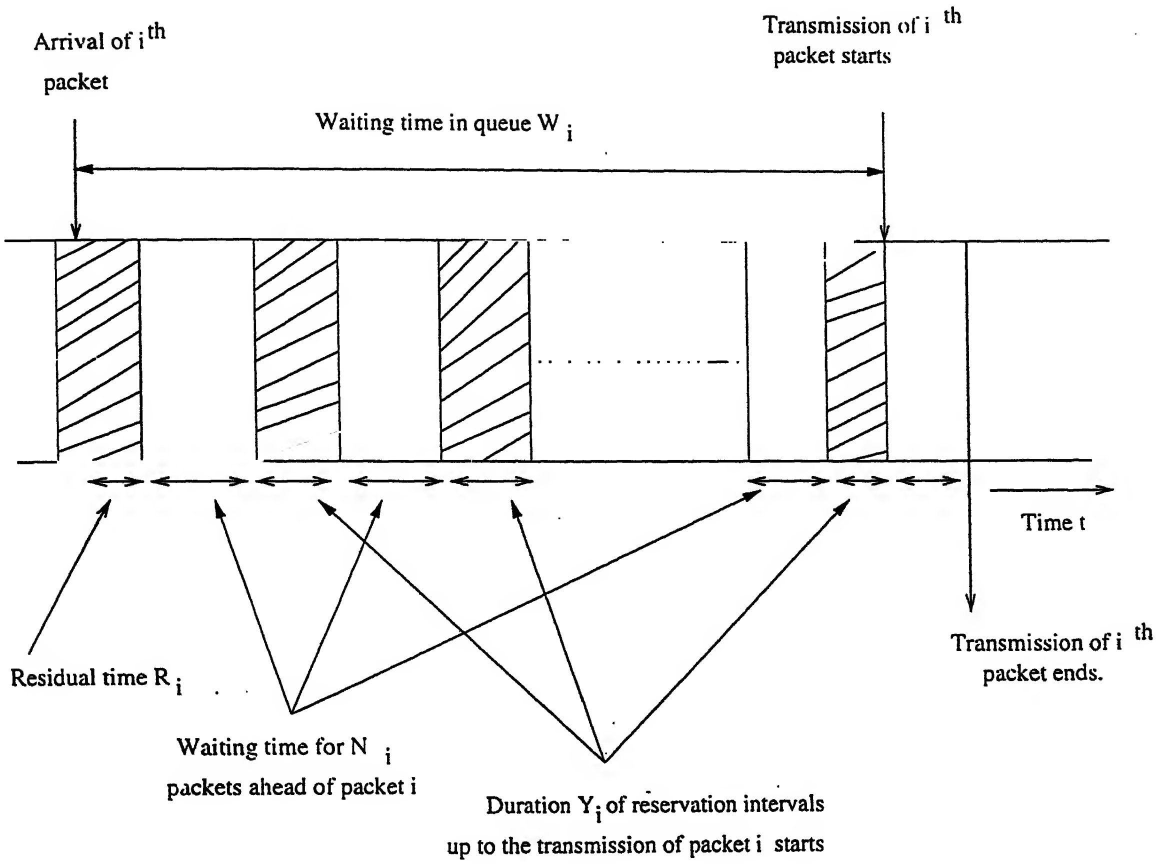


Figure 3.2: Calculation of the average waiting time in the uplink with Round-Robin scheduling

time for each slave's packets are $\overline{X}$ and $\overline{X^2}$, respectively. We denote $\overline{V_i}$ and $\overline{V_i^2}$, respectively, as the first two moments of the polling intervals of slave i. Here note that when a particular slave is polled by the master, only single packet from the slave queue is transmitted. The service times and polling intervals are all independent. We number the slaves 0,1,....,m-1 and assume that l^{th} polling interval is used to poll the slave $l$modm and the subsequent l^{th} data interval is used to send the packets corresponding to that poll. The utilization factor or the offered traffic ρ_u is given by $\lambda_u \overline{X}$

Consider the i^{th} packet arrival in to the system (counting packets in order of arrival, regardless of slave). The expected delay for this packet consists of three terms:

(i) the mean residual time for the packet or polling in progress
(ii) the expected time to transmit the number N_i of packets that must be transmitted before packet i
(iii) the expected duration of polling intervals. See Fig. 3.2.
Thus, the expected queueing delay for the i^{th} packet is given by,

$$E\{W_i\} = E\{R_i\} + E\{N_i\}\overline{X} + E\{Y_i\} \tag{3.1}$$

Taking the limit as $i \to \infty$,

$\lim_{i\to\infty} E\{W_i\} = W_{qu}$, expected time in queue,

$\lim_{i\to\infty} E\{R_i\} = R$, time average mean residual time,

$\lim_{i\to\infty} E\{N_i\}\overline{X} = \rho_u W_{qu}$, by Little's formula,

$\lim_{i\to\infty} E\{Y_i\} = Y$, expected duration of polling intervals.

We can thus write the steady-state version of Eq. 3.1-

$$W_{qu} = R + \rho_u W_{qu} + Y \tag{3.2}$$

The time average mean residual time R, can be calculated using "*Residual Life-Time Approach*".

Residual Life-Time Approach: Fig. 3.3 shows the plot of the *residual life-time r(t)* as a function of *time* t . In this case, the residual life time measures the time left to the end of the current packet service or the current polling, depending on whether a service or a polling is on going at time t. Let X_i be the i^{th} service time and P_j the j^{th} polling interval. Consider the interval (0,t) and evaluate the mean value of r(t) over this interval as:

Let M(t) be the number of services completed by time t, and let L(t) be the number of polling intervals completed by time t.

Time average of r(t) over (0,t) is given by,

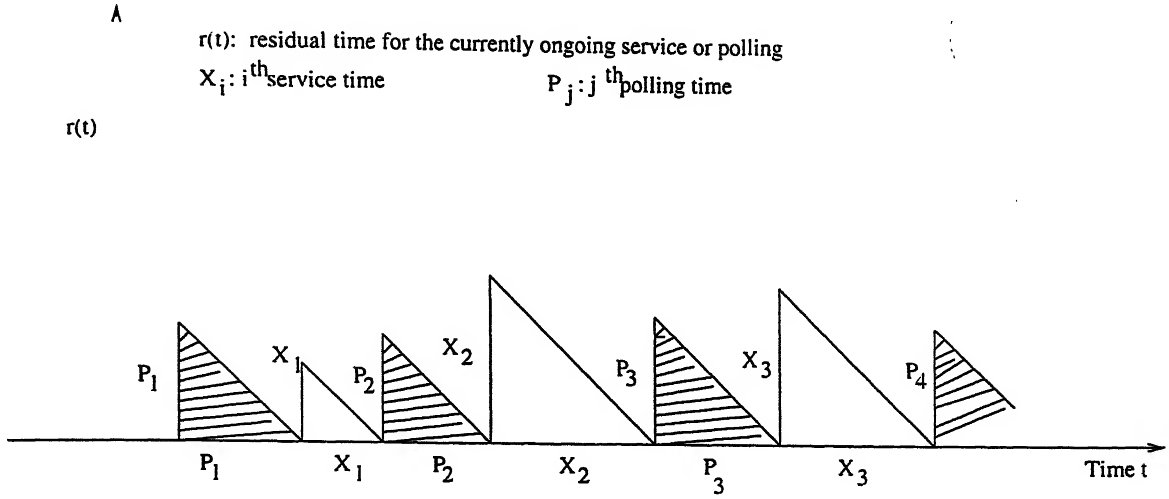


Figure 3.3: Residual life-time r(t) as a function of time

$$\frac{1}{t}\int_0^t r(x)dx = \frac{1}{t}\sum_{i=1}^{M(t)}\frac{1}{2}X_i^2 + \frac{1}{t}\sum_{j=1}^{L(t)}\frac{1}{2}P_j^2$$

$$= \frac{1}{2}\frac{M(t)}{t}\frac{1}{M(t)}\sum_{i=1}^{M(t)} X_i^2 + \frac{1}{2}\frac{L(t)}{t}\frac{1}{L(t)}\sum_{i=1}^{L(t)} P_j^2 \tag{3.3}$$

Taking limits of the term above as $t \to \infty$, we will get-

$$R = \lim_{t\to\infty}\frac{1}{t}\int_0^t r(x)dx; \qquad \lim_{t\to\infty}\frac{M(t)}{t} = \lambda_u$$

Assuming that time averages can be replaced by ensemble averages, we have

$$\lim_{t\to\infty}\frac{1}{M(t)}\sum_{i=1}^{M(t)} X_i^2 = \overline{X^2}; \qquad \lim_{t\to\infty}\frac{1}{L(t)}\sum_{j=1}^{L(t)} P_j^2 = \sum_{l=0}^{m-1}\overline{V_l^2}$$

As time $t \to \infty$, the fraction of time spent serving the packets in the system approaches ρ_u and thus the fraction of time occupied with polling intervals is $(1-\rho_u)$. Then we have,

$$\lim_{t\to\infty}\frac{t(1-\rho_u)}{L(t)} = \sum_{l=0}^{m-1}\overline{V_l}$$

or

$$\lim_{t\to\infty} \frac{L(t)}{t} = \frac{(1-\rho_u)}{\sum_{l=0}^{m-1} \overline{V_l}}$$

Using the above in Eq. 3.3, the time average mean residual time (R) is given by-

$$R = \frac{1}{2}\lambda_u \overline{X^2} + \frac{1}{2}(1-\rho_u)\frac{\sum_{l=0}^{m-1} \overline{V_l^2}}{\sum_{l=0}^{m-1} \overline{V_l}} \tag{3.4}$$

Whereas the term Y, mean duration of polling intervals, is given by

$$Y = \frac{(m-\rho_u)\overline{V}}{2} - \frac{(1-\rho_u)\sum_{l=0}^{m-1} \overline{V_l^2}}{2m\overline{V}} + \lambda_u W_{qu} \overline{V} \qquad (See Appendix B.1) \tag{3.5}$$

$$where \qquad \overline{V} = \frac{1}{m}\sum_{l=0}^{m-1} \overline{V_l}$$

Combining Eqs. 3.2, 3.4, and 3.5, we obtain

$$W_{qu} = \frac{\lambda_u \overline{X^2}}{2(1-\rho_u-\lambda_u\overline{V})} + \frac{(m+\rho_u)\overline{V}}{2(1-\rho_u-\lambda_u\overline{V})} + \frac{\sigma_V^2(1-\rho_u)}{2\overline{V}(1-\rho_u-\lambda_u\overline{V})} \tag{3.6}$$

$$where \qquad \overline{V} = \frac{1}{m}\sum_{l=0}^{m-1} \overline{V_l}$$

$$\sigma_V^2 = \frac{\sum_{l=0}^{m-1}(\overline{V_l^2} - \overline{V_l}^2)}{m}$$

Therefore, mean delay for a packet in the uplink is given by

$$W_u = W_{qu} + \overline{X} \tag{3.7}$$

3.1.2 Downlink

In a piconet, for each slave there is a corresponding queue at the master. Master sends a single packet from each queue in a cyclic (round-robin) order. After the master has completed sending a packet from a queue and before it begins work on next queue there is a period during which there is a receiving time from the

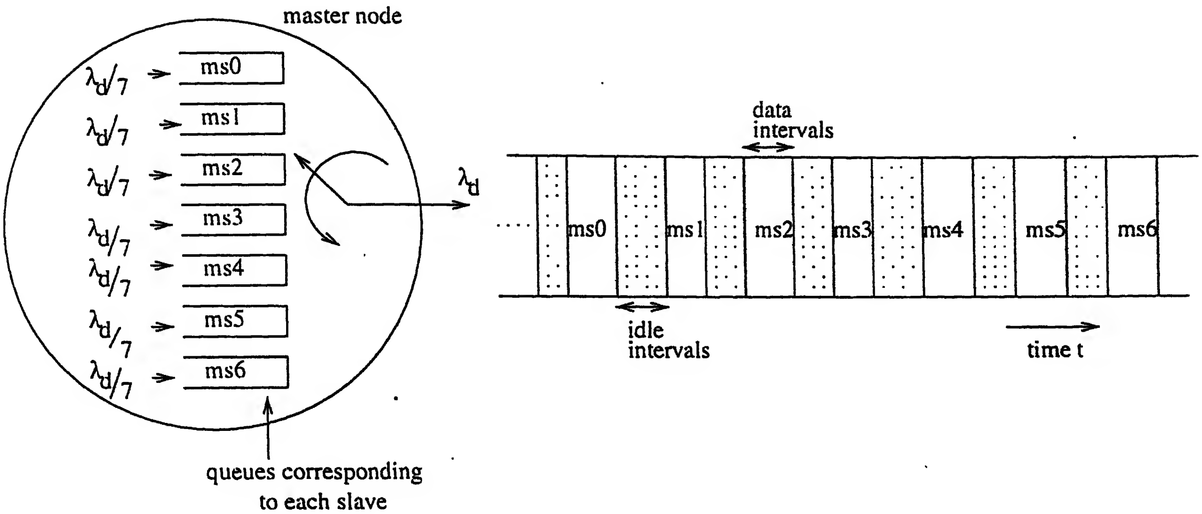


Figure 3.4: Illustration of channel in downlink with Round-Robin scheduling

corresponding slave. These receiving periods are the idle periods for the queues present at the master. This model is depicted in Fig. 3.4.

In the downlink, the time axis is devided into *data intervals*, where actual data from the queues of the master is transmitted, and *idle intervals* corresponding to the data transmission intervals from slaves. Each *data interval* is preceded by an *idle interval.*

We will consider m (seven) traffic streams (packets arriving each queue of the master) corresponding to each slave. The arrival process at each queue is assumed to be Poisson with an arrival rate of λ_d/m. The first and second moments of the service time of a packet are $\overline{X}$ and $\overline{X^2}$, respectively. Here note that in each data interval only a single packet from a single queue will be transmitted. We denote $\overline{I_i}$ and $\overline{I_i^2}$ respectively, the first and second moments of the idle intervals preceding the data intervals of queue i. The service times and idle intervals are all independent. We number the queues of master as 0,1,.....,(m-1), and assume that l^{th} idle interval is the preceding interval of l^{th} data interval corresponding to queue lmodm. The utilization factor or offered traffic is ρ_d is given by $\lambda_d\overline{X}$.

Consider i^{th} packet arrival into the system (counting packets in the order of arrival, regardless of queue). The expected delay for this packet consists of three terms:

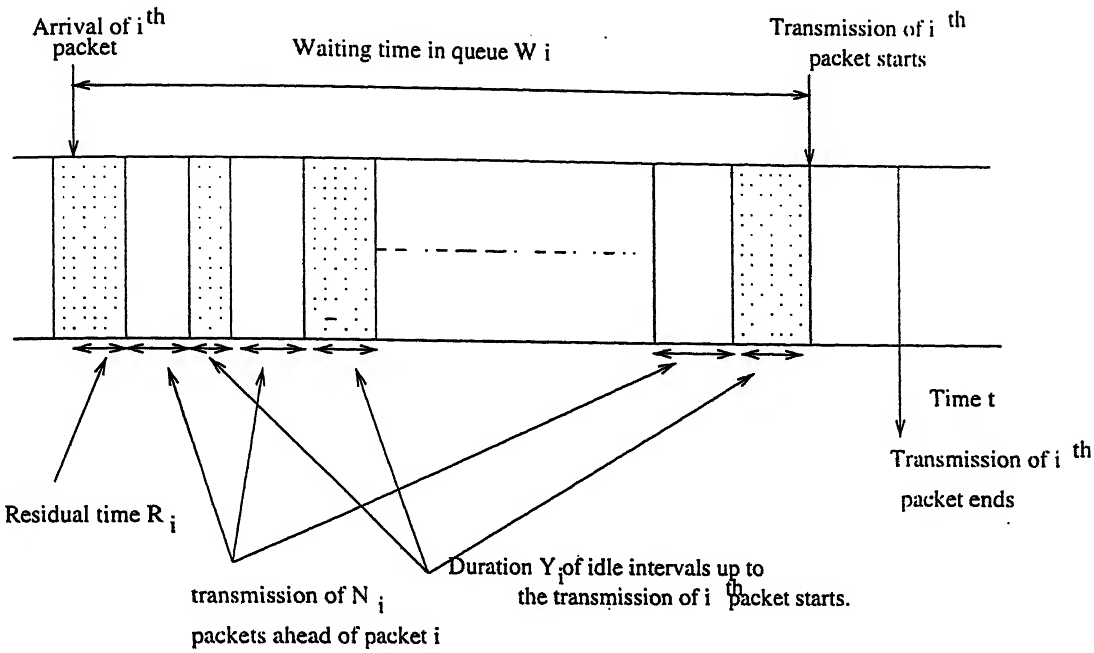


Figure 3.5: Calculation of average waiting time in the downlink with round-robin scheduling

(i) the mean residual time for the packet transmission interval or the idle interval in progress
(ii) the expected time to transmit the number N_i of packets that must be transmitted before packet i
(iii) the expected duration of idle intervals. See Fig. 3.5.

Thus, the expected queueing delay for the i^{th} packet is given by,

$$E\{W_i\} = E\{R_i\} + E\{N_i\}\overline{X} + E\{Y_i\} \tag{3.8}$$

Taking the limit as $i \to \infty$,

$\lim_{i\to\infty} E\{W_i\} = W_{qd}$, expected time in queue,

$\lim_{i\to\infty} E\{R_i\} = R$, time average mean residual time,

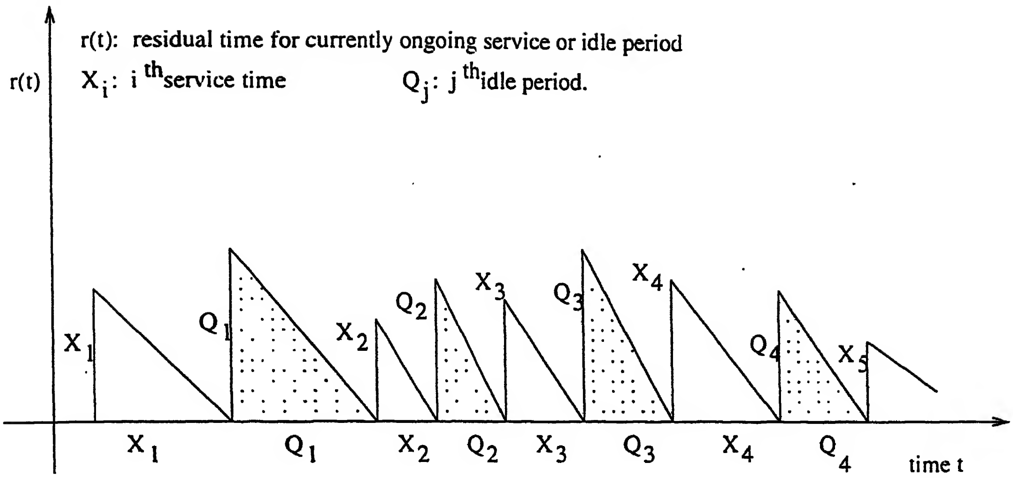


Figure 3.6: Residual life-time as a function of time

$\lim_{i\to\infty} E\{N_i\}\overline{X} = \rho_d W_{qd}$, by Little's formula,

$\lim_{i\to\infty} E\{Y_i\} = Y$, expected duration of idle intervals.

We can thus write the steady-state version of Eq. 3.8-

$$W_{qd} = R + \rho_d W_{qd} + Y \tag{3.9}$$

The time average mean residual time R, can be calculated using "*Residual Life-Time Approach*".

Residual Life-Time Approach: Fig. 3.6 shows the plot of the *residual life-time r(t)* as a function of ***time*** t .

In this case, the residual life time measures the time left to the end of the current packet service or the current idle period, depending on whether a service or a idle period is on going at time t. Let X_i be the i^{th} service time and Q_j the j^{th} idle period. Consider the interval (0,t) and evaluate the mean value of r(t) over this interval as:

Let M(t) be the number of services completed by time t, and let L(t) be the number of idle periods completed by time t.

Time average of r(t) over (0,t) is given by,

$$\frac{1}{t}\int_0^t r(x)dx = \frac{1}{t}\sum_{i=1}^{M(t)} \frac{1}{2}X_i^2 + \frac{1}{t}\sum_{j=1}^{L(t)} \frac{1}{2}Q_j^2$$

$$= \frac{1}{2}\frac{M(t)}{t}\frac{1}{M(t)}\sum_{i=1}^{M(t)} X_i^2 + \frac{1}{2}\frac{L(t)}{t}\frac{1}{L(t)}\sum_{i=1}^{L(t)} Q_j^2 \tag{3.10}$$

Taking limits of the term above as $t \to \infty$, we will get-

$$R = \lim_{t\to\infty} \frac{1}{t}\int_0^t r(x)dx; \qquad \lim_{t\to\infty} \frac{M(t)}{t} = \lambda_d$$

Assuming that time averages can be replaced by ensemble averages, we have

$$\lim_{t\to\infty} \frac{1}{M(t)}\sum_{i=1}^{M(t)} X_i^2 = \overline{X^2}; \qquad \lim_{t\to\infty} \frac{1}{L(t)}\sum_{j=1}^{L(t)} Q_j^2 = \sum_{l=0}^{m-1} \overline{I_l^2}$$

As time $t \to \infty$, the fraction of time spent serving the packets in the system approaches ρ_d and thus the fraction of time occupied with idle intervals is $(1-\rho_d)$. Then we have,

$$\lim_{t\to\infty} \frac{t(1-\rho_d)}{L(t)} = \sum_{l=0}^{m-1} \overline{I_l}$$

or

$$\lim_{t\to\infty} \frac{L(t)}{t} = \frac{(1-\rho_d)}{\sum_{l=0}^{m-1} \overline{I_l}}$$

Using the above in Eq. 3.10, the time average mean residual time (R) in this case is given by-

$$R = \frac{1}{2}\lambda_d\overline{X^2} + \frac{1}{2}(1-\rho_d)\frac{\sum_{l=0}^{m-1} \overline{I_l^2}}{\sum_{l=0}^{m-1} \overline{I_l}} \tag{3.11}$$

Whereas the term Y, mean duration of idle intervals, is given by

$$Y = \frac{(m-\rho_d)\overline{I}}{2} - \frac{(1-\rho_d)\sum_{l=0}^{m-1} \overline{I_l^2}}{2m\overline{I}} + \lambda_d W_q \overline{I} \qquad (See Appendix B.2) \tag{3.12}$$

$$where \qquad \overline{I} = \frac{1}{m}\sum_{l=0}^{m-1} \overline{I_l}$$

Combining Eqs. 3.9, 3.11, and 3.12, we obtain

$$W_{qd} = \frac{\lambda_d \overline{X^2}}{2(1-\rho_d-\lambda_d\overline{I})} + \frac{(m+\rho_d)\overline{I}}{2(1-\rho_d-\lambda_d\overline{I})} + \frac{\sigma_I^2(1-\rho_d)}{2\overline{I}(1-\rho_d-\lambda_d\overline{I})} \tag{3.13}$$

$$where \qquad \overline{I} = \frac{1}{m}\sum_{l=0}^{m-1}\overline{I_l}$$

$$\sigma_I^2 = \frac{\sum_{l=0}^{m-1}(\overline{I_l^2} - \overline{I_l}^2)}{m}$$

Therefore, mean delay for a packet in the downlink is given by

$$W_d = W_{qd} + \overline{X} \tag{3.14}$$

3.2 Classification of traffic

In a communication system, there will be various kinds of traffic with different QoS (Quality-of-Service) requirements. So it is important to support various traffic with QoS gaurantees. Broadly, diverse traffic may be categorized as real-time traffic and non-real-time traffic. Bluetooth supports both real-time and non-real-time communication services. Since we are dealing with non-real-time traffic, depending on its distinct characteristics and QoS requirements, the non-real-time traffic can be divided into two classes: a) class1: delay-tolerant traffic like paging and email; and b) class2: delay sensitive traffic like FTP and remote log-in. Class2 should be given priority over class1. The main distinguishing factor between these classes is how delay sensitive the traffic is. Class2 can also be called as interactive class mainly used by interactive applications whereas class1 can be called as Background class which is meant for background traffic.

To satisfy the QoS needs of different connections, nodes present in the network need to have priority and scheduling mechanisms. A simple master driven round-robin scheduling in Bluetooth is unable to minimize delay for interactive sessions. So in the following sections we derive a new methods to support two classes of service (class1 and class2) in Bluetooth through priority and scheduling mechanisms. In our proposed methods, class2 traffic is given priority over class1. The two important parameters that have been considered are minimizing end-to-end packet delivery delay and providing consistent data throughput and capacity.

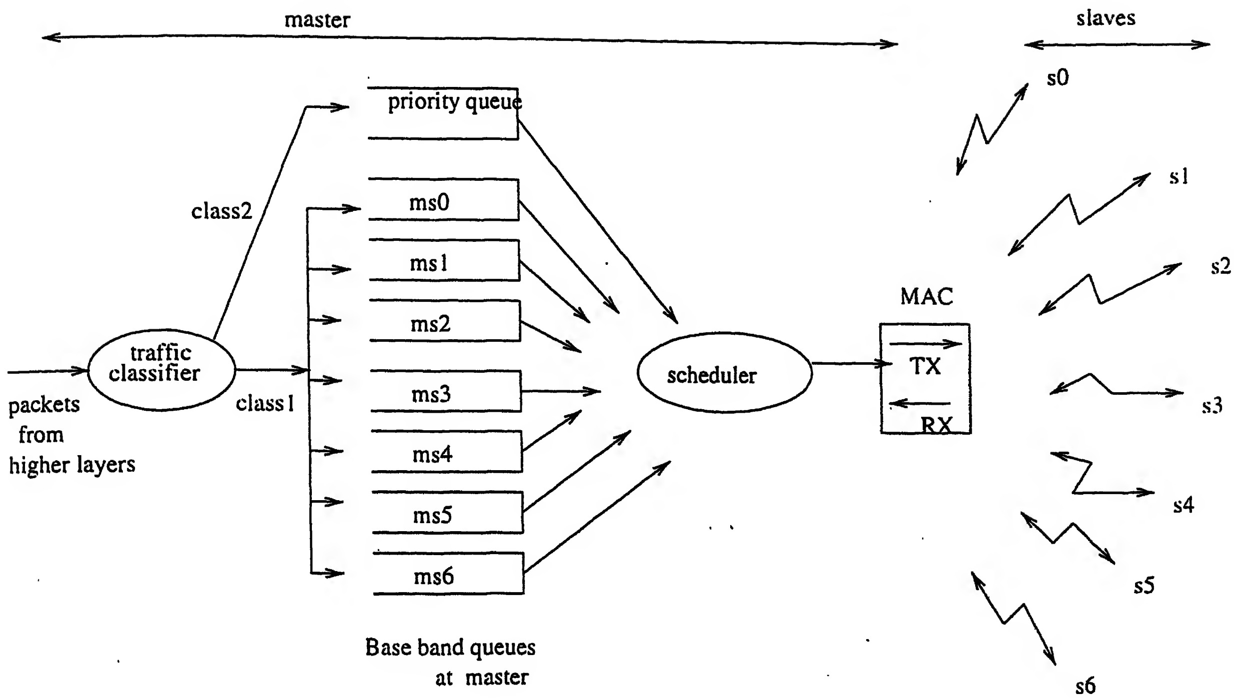


Figure 3.7: Illustration of downlink with a 'Priority Queueing (PQ) at master' model

3.3 Priority Queueing (PQ) at master

We consider a piconet, as it is taken in section 3.1, consisting of one master and seven slaves. In a piconet master could be a fixed access point or a base station. In this priority queueing (PQ) model master maintains one priority queue for a class2 traffic, for all the slaves, and an individual output class1 queues for each slave. Before queueing the packets, the traffic classifier devides each packet into either class2 or class1. After classification, the packets will be queued into the respective queues. This model is depicted in Fig. 3.7.

With this priority queueing model, the scheduling of packets from different queues at master on to an output channel is as follows: When the master choosing a packet to transmit in each master-to-slave slot, the priority queueing descipline will transmit a packet from the highest priority class (class2) that has a non-empty queue (i.e., has packets waiting for transmission). Master keeps on serving the class2

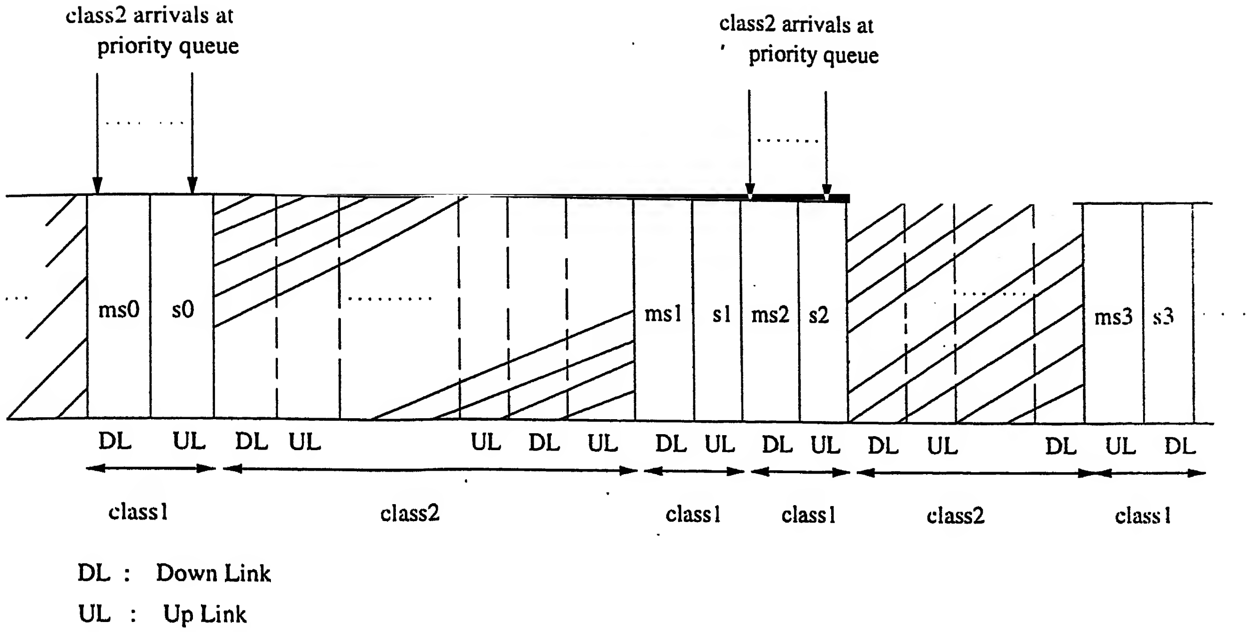


Figure 3.8: Illustration of channel in the downlink with 'Priority Queueing(PQ) at master' model

packets in a FIFO (First-In-First-Out) manner until the class2 priority queue becomes empty. So the service process at the class2 priority queue is an exhaustive service process. Master will transmit a packet from class1 queue only if the class2 queue becomes empty. It serves the class1 queues in round-robin manner and the service process is limited-1 service. While the master serving the packet from class1 queue, if a class2 packet has arrived then the master follows a non-preemptive priority queueing policy. Because it is not possible for preemptive priority queueing policies in Bluetooth. Under a so-called non-preemptive priority queueing discipline, the transmission of a packet is not interrupted once it has begun. When the master sends a packet, it uses the MAC address of slave in the packet header as each slave will have a unique MAC address in a piconet. After receiving a packet from the master, the corresponding slave will send a packet to the master in the immediate slave-to-master slot. This kind of scheduling is shown in Fig. 3.8.

In the priority queueing discipline, the mean delay for class2 packets can be calculated analytically by modeling the class2 priority queue as "*M/G/1 queue with vacations*" [12], if we assume the arrival process of class2 packets into the priority

queue as Poisson and vacation intervals as the intervals during which master spends at a single class1 queue.

M/G/1 queue with vacations: Consider M/G/1 queue, where after each busy period, the server goes on a vacation for a random interval of time. Arrivals during the vacation go into service after the server returns from vacation. If on returning from a vacation, the server finds the queue empty, then it goes on another vacation of random length. This continues until the server returns from the vacation and finds packets waiting in the queue. After it starts service, following a vacation, the server continues serving normally (like a normal M/G/1 queue) until the system becomes empty once again.

Similarly in our model, master keeps on serving the class2 priority queue until it becomes empty. After the priority queue becomes empty, master serves the class1 queue transmitting a class1 packet. Once the pair of slots corresponding to one class1 queue are completed, if the master finds the priority queue empty, then master serves another class1 queue (in round-robin manner). This process continues until the priority queue is non-empty. After master starts serving priority queue, it continues until it becomes empty.

The mean delay of a class2 packet can be calculated using "*Residual Life-Time Approach*".

Residual Life-Time Approach: We assume that arrival process of class1 and class2 packets is Poisson with an effective arrival rates λ_{2d} and λ_{1d}, respectively. Here the service time of a class2(class1) packet is defined as the effective service time during which master sends a class2(class1) packet in master-to-slave slot and receives acknowledgement from the corresponding slave in slave-to-master slot. So the service time corresponds to duration of pair of slots which is random. Let $\overline{X_e}$ and $\overline{X_e^2}$ are the first and second moments, respectively of the service time. The vacation peroid is the service time of a class1 packet. Let $\overline{V_e}$ and $\overline{V_e^2}$ are the first and second moments, respectively of the vacation period.

For a class2 priority queue, consider a class2 packet arriving into the queue. The mean queueing delay for this packet consists of two terms:

(i) the mean residual time for the service or the vacation period

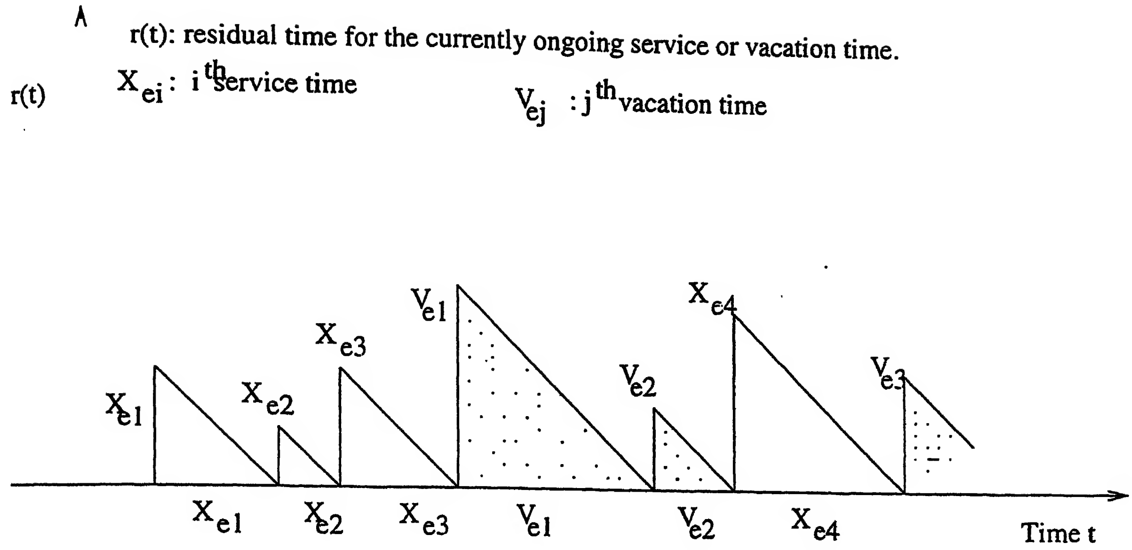


Figure 3.9: Residual life-time as a function of time

(ii) the expected time to transmit the number N_{2qd} of packets that must be transmitted before the arrival of interest.

So the mean queueing delay is given by-

$$W_{2qd} = \overline{N}_{2qd}\overline{X_e} + R \tag{3.15}$$

From Little's theorem, $\overline{N}_{2qd} = \lambda_{2d} W_{2qd}$.

Therefore, we can write-

$$W_{2qd} = \lambda_{2d} W_{2qd}\overline{X_e} + R \tag{3.16}$$

Defining $\rho_{2d} = \lambda_{2d}\overline{X_e}$, we get

$$W_{2qd} = \frac{R}{(1 - \rho_{2d})} \tag{3.17}$$

To find W_{2qd}, we would need the mean residual time R, which may be found using a graphical approach. Fig. 3.9 shows the plot of the *residual life-time r(t)* as a function of *time* t .

In this case, the resudual life time measures the time left to the end of the current packet service or the current vacation period, depending on whether a service or a

vacation period is ongoing at time t. Let X_{ei} be the i^{th} service time and V_{ej} the j^{th} vacation period. Consider the interval (0,t) and evaluate the mean value of r(t) over this interval as:

Let M(t) be the number of services completed by time t, and let L(t) be the number of vacation periods completed by time t.

Time average of r(t) over (0,t) is given by,

$$\frac{1}{t}\int_0^t r(x)dx = \frac{1}{t}\sum_{i=1}^{M(t)}\frac{1}{2}X_{ei}^2 + \frac{1}{t}\sum_{j=1}^{L(t)}\frac{1}{2}V_{ej}^2$$

$$= \frac{1}{2}\frac{M(t)}{t}\frac{1}{M(t)}\sum_{i=1}^{M(t)}X_{ei}^2 + \frac{1}{2}\frac{L(t)}{t}\frac{1}{L(t)}\sum_{i=1}^{L(t)}V_{ej}^2 \tag{3.18}$$

Taking limits of the term above as $t \to \infty$, we will get-

$$R = \lim_{t\to\infty}\frac{1}{t}\int_0^t r(x)dx; \qquad \lim_{t\to\infty}\frac{M(t)}{t} = \lambda_{2d}$$

Assuming that time averages can be replaced by ensemble averages, we have

$$\lim_{t\to\infty}\frac{1}{M(t)}\sum_{i=1}^{M(t)}X_i^2 = \overline{X_e^2}; \qquad \lim_{t\to\infty}\frac{1}{L(t)}\sum_{j=1}^{L(t)}V_j^2 = \overline{V_e^2}$$

As time $t \to \infty$, the fraction of time spent serving the packets in the system approaches ρ_{2d} and thus the fraction of time occupied with vacation intervals is $(1-\rho_{2d})$. Then we have,

$$\lim_{t\to\infty}\frac{t(1-\rho_{2d})}{L(t)} = \overline{V_e}$$

or

$$\lim_{t\to\infty}\frac{L(t)}{t} = \frac{(1-\rho_{2d})}{\overline{V_e}}$$

Using the above in Eq. 3.18, the time average mean residual time (R) in this case is given by-

$$R = \frac{1}{2}\lambda_{2d}\overline{X_e^2} + \frac{1}{2}(1-\rho_{2d})\frac{\overline{V_e^2}}{\overline{V_e}} \tag{3.19}$$

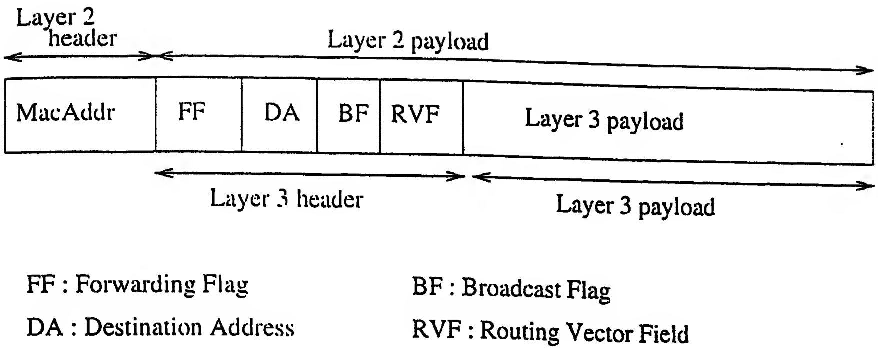


Figure 3.10: Relevant fields in a packet for routing in Bluetooth

Using the above in 3.17, we get the mean queueing delay of class2 packets in a priority queue as-

$$W_{2qd} = \frac{\lambda_{2d}\overline{X_e^2}}{2(1-\rho_{2d})} + \frac{\overline{V_e^2}}{2\overline{V_e}} \tag{3.20}$$

This analysis was based on the assumption that the intervals X_{ei} and V_{ei} are independent of each other and also independent of the arrival process.

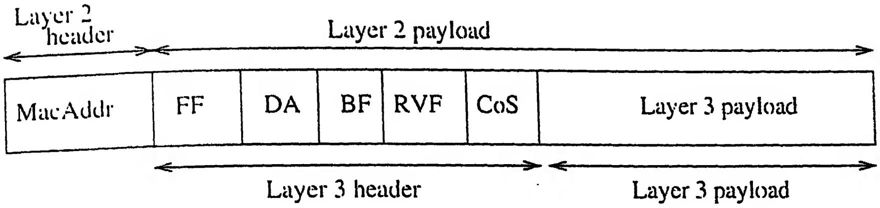
Therefore, the mean delay for class2 packets with a priority queue is given by-

$$W_{2d} = W_{2qd} + \overline{X} \tag{3.21}$$

3.4 Priority queueing at master with priority polling

In this section, we consider a piconet composed of a master and seven slaves, where slaves are sending packets to other slaves in the piconet. Here master acts as a router receiving the packets from slaves and forwarding them to the destination nodes. Since Bluetooth inherently doesn't support direct slave-to-slave connectivity at the link or radio link level. it will necessarily be done by routing via the designated master for the piconet. The routing algorithm proposed for Bluetooth in [6] uses Layer 3 control information, which will be incorporated in the Layer 3 header (see Fig. 3.10), for intra and inter piconet communication.

The Layer 3 header in the Layer 2 payload of packet will contain a forwarding flag(FF) and a destination MacAddr(DA) field. If FF=1, then the payload of the



FF : Forwarding Flag

BF : Broadcast Flag

DA : Destination Address

RVF : Routing Vector Field

CoS : Class of Service

Figure 3.11: A packet format in Bluetooth with added CoS bit to provide classes of service in Bluetooth

packet is intended for another slave unit in the same piconet and DA contains the MacAddr of the destination unit. The MacAddr present in the Layer 2 header corresponds to the MacAddr of the sending slave. When the master receives a packet. it strips the Layer2 header and forwards to the Layer 3. If FF=1, then the Layer 3 processor puts the payload in a new packet and sends it to the corresponding slave. For example the payload of a packet received by the master with FF=1, DA=010. MacAddr = 101 will be forwarded by the master to the slave with MacAddr=010.

A packet in slave-to-slave communication may be of class2 type packet or class1 type. To provide different classes of service, there is a need for explicit marking of packet's priority class in the packet header. So we have added one more bit called "Class of Service(CoS)" bit in the Layer 3 header (see Fig 3.11). If CoS=1, the payload of the packet is of class2 type. The value CoS=0 is meant for class1.

In this "Priority queueing at master with priority polling" model. we have devided the slaves into two groups. First group consists of 'p' number of slaves out of seven slaves which are generating class2 type of data. Second group consists of remaining slaves generating class1 kind of data. A packet generated by a slave is destined to any of other slave according to uniform distribution. When a master receives a packet. if CoS=1 then the packet will be queued in class2 priority queue at the master node. If CoS=0, then the packet will be queued in respective destination class1 output queue at master. In the following, we have derived the mean

end-to-end delays for class2 packets by analytical modeling when p=1. Later it is generalized for p greater than 1.

Since the master schedules the traffic in both the uplink and downlink, the new scheduling algorithm derived here will take into account the slave characteristics. When a master polling the slaves, the polling will be done in two ways: exhaustive polling and limted-1 polling. Exhaustive polling is done for slaves generating class2 traffic, i.e., once master started polling the slave, it continuously polls the same slave until queue of the slave becomes zero. Whereas limited-1 polling is for slaves generating class1 traffic, i.e., master polls the slave to receive only the first packet from that slave's queue. Here the polling refers to either explicit polling or implicit polling.

The new scheduling algorithm is explained here for p=1. Master always gives priority to class2 slave in polling the slaves. It polls the class1 slave only if there is no backlog at class2 slave. As it is mentioned earlier, master implements exhaustive polling for class2 slaves. When the master polling the class2 source exhaustively, all the packets master has received will be queued in class2 priority queue at the master. If the queue of class2 source becomes zero then the master stops polling the source and starts sending all the packets, that are received during polling, to the destinations. After the priority queue at master becomes zero, master polls the class2 source only if it is backlogged. If it is not backlogged, master polls the class1 slaves in round-robin manner. In this model, we assume that the binary information regarding the status of the queue at the slave is available at the master. Because a "Predictive Fair Poller", which predicts the status of slave queue, has been proposed in [13].

In this model, the arrival process at the slave queues is assumed as Poisson. Then, the queue of the class2 source can be modeled as a M/G/1 queue with vacations in which first vacation period is exceptional than the subsequent vacation periods, i.e., the first vacation period has a different distribution than the subsequent vacation's distribution. Here, the first vacation period corresponds to the period during which master sends the packets from it's priority queue to the destinations. The subsequent vacation periods are the periods during which master polls a class1 slave and receives

a packet from the corresponding slave. The mean end-to-end delays for class2 packets can be calculated using again "*Residual Life-time Approach*".

Residual Life-Time Approach: We assume that the arrival process of packets at class2 and class1 slave queues is Poisson with an effective arrival rates λ_2 and λ_1 , respectively. Here the service time of a class2 (class1) packet is defined as the effective service time during which master polls the class2 (class1) slave in master-to-slave slot and receives a packet from the corresponding slave in the slave-to-master slot. So the service time corresponds to the duration of a pair of master-to-slave and slave-to-master slots which is random. Let $\overline{X_e}$ and $\overline{X_e^2}$ are the first and second moments, respectively of the service time. The first vacation period is equal to the length of the "Busy Period" of class2 slave's queue. Let $\overline{B}$ and $\overline{B^2}$ are the first and second moments of the Busy Period of class2 slave's queue. The subsequent vacation periods are equal to the service times of class1 packets. Let $\overline{V_e}$ and $\overline{V_e^2}$ are the first and second moments, respectively of those vacation periods.

At a class2 slave, consider a class2 packet arriving into the queue. The mean queueing delay for this packet consists of two terms.
(i) the mean residual time for the service or the vacation period.
(ii) the expected time to transmit the number N_{2q} of packets that must be transmitted before arrival of interest.

So the mean queueing delay of class2 packet at class2 slave's queue is given by-

$$W_{2q} = \overline{N_{2q}X_e} + R \tag{3.22}$$

From Little's theorem, $\overline{N_{2q}} = \lambda_2 W_{2q}$
Therefore we can write-

$$W_{2q} = \lambda_2 W_{2q}\overline{X_e} + R \tag{3.23}$$

Defining $\rho_2 = \lambda_2 \overline{X_e}$, we get

$$W_{2q} = \frac{R}{(1-\rho_2)} \tag{3.24}$$

To find W_{2q}, we would need the mean residual time R, which may be found using a graphical approach. Fig. 3.12 shows the plot of the *residual life-time r(t)* as a

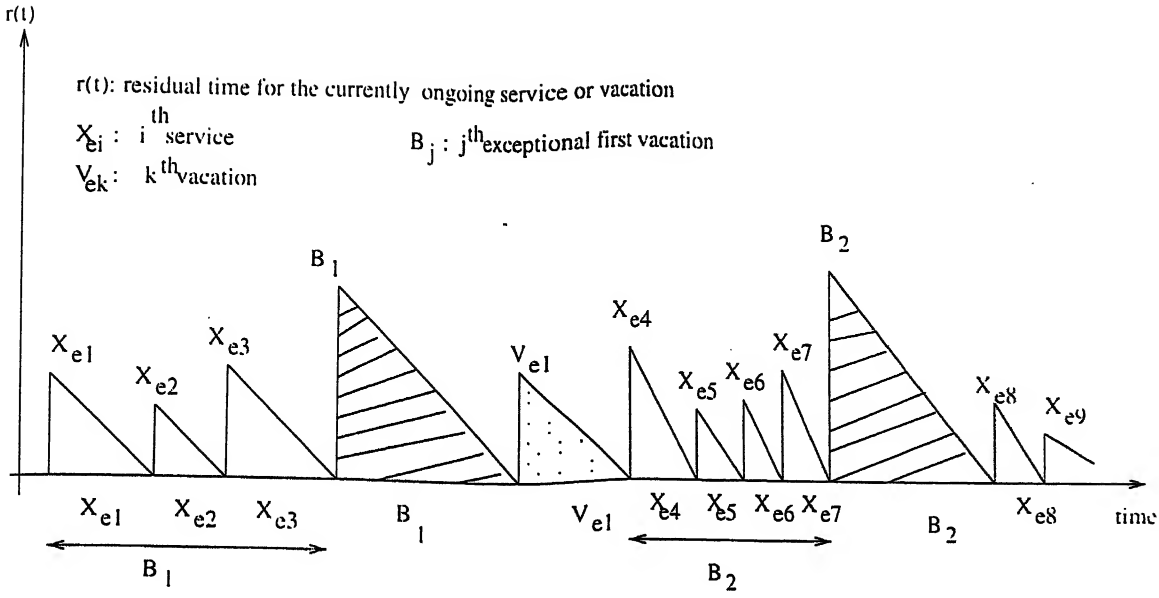


Figure 3.12: Residual life-time as a function of time

function of *time* t . In this case, the residual life time measures the time left to the end of the current packet service or the current vacation period, depending on whether a service or a vacation period is on going at time t. Let X_{ei} be the i^{th} service time, B_j be the j^{th} vacation period equal to busy period of class2 queue. V_{ek} the k^{th} vacation period equal to the service time class1 packet.

Consider the interval (0,t) and evaluate the mean value of r(t) over this interval as:

Let M(t) be the number of services completed by time t, and let N(t) be the number of vacation periods equal to the busy periods of class2 queue, K(t) be the number of vacation periods equal to the service times of class1 packet completed by time t.

Time average of r(t) over (0,t) is given by,

$$\frac{1}{t}\int_0^t r(x)dx = \frac{1}{t}\sum_{i=1}^{M(t)}\frac{1}{2}X_{ei}^2 + \frac{1}{t}\sum_{j=1}^{N(t)}\frac{1}{2}B_j^2 + \frac{1}{t}\sum_{k=1}^{K(t)}\frac{1}{2}V_{ej}^2$$

$$= \frac{1}{2}\frac{M(t)}{t}\frac{1}{M(t)}\sum_{i=1}^{M(t)} X_{ei}^2 + \frac{1}{2}\frac{N(t)}{t}\frac{1}{N(t)}\sum_{j=1}^{N(t)} B_j^2 + \frac{1}{2}\frac{K(t)}{t}\frac{1}{K(t)}\sum_{k=1}^{K(t)} V_{ej}^2 \qquad (3.25)$$

Taking limits of the term above as $t \to \infty$, we will get-

$$R = \lim_{t\to\infty} \frac{1}{t}\int_0^t r(x)dx; \qquad \lim_{t\to\infty} \frac{M(t)}{t} = \lambda_2$$

Assuming that time averages can be replaced by ensemble averages, we have

$$\lim_{t\to\infty} \frac{1}{M(t)} \sum_{i=1}^{M(t)} X_i^2 = \overline{X_e^2}; \qquad \lim_{t\to\infty} \frac{1}{N(t)} \sum_{j=1}^{N(t)} B_j^2 = \overline{B^2}; \qquad \lim_{t\to\infty} \frac{1}{K(t)} \sum_{k=1}^{K(t)} V_k^2 = \overline{V_e^2}$$

As time $t \to \infty$, the fraction of time spent serving the packets of class2 source approaches ρ_2 and thus the fraction of time occupied with vacation intervals is $(1-\rho_2)$. Then we have,

$$N(t)\overline{B} + N(t)\overline{B} + K(t)\overline{V_e} = t \tag{3.26}$$

or

$$2N(t)\overline{B} + K(t)\overline{V_e} = t \tag{3.27}$$

Also, we have

$$N(t)\overline{B} + K(t)\overline{V_e} = t(1-\rho_2) \tag{3.28}$$

From Eqs. 3.27, 3.28, we will obtain-

$$\frac{2N(t)}{t}\overline{B} + \frac{K(t)}{t}\overline{V_e} = 1 \tag{3.29}$$

$$\frac{N(t)}{t}\overline{B} + \frac{K(t)}{t}\overline{V_e} = (1-\rho_2) \tag{3.30}$$

Solving Eqs. 3.29, 3.30, we obtain,

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{\rho_2}{\overline{B}}$$

$$\lim_{t\to\infty} \frac{K(t)}{t} = \frac{(1-2\rho_2)}{\overline{V_e}}$$

Using the above in Eq. 3.25, the time average mean residual time is given by-

$$R=\frac{1}{2}\lambda_2\overline{X_e^2}+\frac{1}{2}\rho_2\frac{\overline{B^2}}{\overline{B}}+\frac{1}{2}(1-2\rho_2)\frac{\overline{V_e^2}}{\overline{V_e}} \tag{3.31}$$

From Eqs. 3.24, 3.31, the mean queueing delay of a class2 packet at class2 slave is given by-

$$W_{2q}=\frac{1}{2}\frac{\lambda_2}{(1-\rho_2)}\overline{X_e^2}+\frac{1}{2}\frac{\rho_2}{(1-\rho_2)}\frac{\overline{B^2}}{\overline{B}}+\frac{1}{2}\frac{(1-2\rho_2)}{(1-\rho_2)}\frac{\overline{V_e^2}}{\overline{V_e}} \tag{3.32}$$

The mean queueing delay of a class2 packet at the priority queue of the master is equal to $(\overline{B}-\overline{X_e})$

Where,

$$\rho_2=\lambda_2\overline{X_e}$$

$$\overline{B}=\frac{\overline{X_e}}{(1-\rho_2)} \quad (See Appendix C.1)$$

$$\overline{B^2}=\frac{\overline{X_e^2}}{(1-\rho_2)^3} \quad (See Appendix C.1)$$

Therefore, the mean end-to-end delay for class2 packets is given by

$$W_2=W_{2q}+(\overline{B}-\overline{X_e})+2\overline{X} \tag{3.33}$$

In general, for p greater than one: master polls a backlogged class2 source in an exhaustive manner, i.e., until the queue of the source becomes zero, and receives the packets destined for other nodes in the piconet. Once the class2 source is unbacklogged, it stops polling and forwards all the packets, that are received, to the destinations. With the same process, master serves all the class2 sources present in piconet in a round-robin manner. When the master forwarding the packets of a particular class2 source, it may receive some class2 packets from destination nodes in the slave-to-master slots. These packets will be queued in the priority queue. Master will start polling the next class2 source, only if all the packets in the priority queue are cleared. Also, when master polling a class2 source, it can send a class1 packet destined for that source in the master-to-slave slot. This model can be explained logically as follows.

Consider, two logical states 0 and 1 for class2 sources and master. Class2 source remains in state 0 as long as it is unbacklogged. Master is also in state 0, if it's priority queue is empty. A class2 source will enter into state 1, if it is backlogged. The master will also enter into state 1 if the priority queue is non-empty. Master maintains a separate counter corresponding to each class2 source whose values are also can be either zero or one. Initially assume that master is at state 0, since it only acts as router, and all the counters are also in zero state.

Step1: Now the master polls the class2 source whose state is 1 until source enters into state 0. Once master started polling, the counter value corresponding to that source becomes 1.

Step2: Then the master will enter into state 1.

Step3: Master keeps on sending the packets that are arrived at it's priority queue until it enters into state 0.

Step4: Once the master enters into state 0, it polls the class2 source whose state is 1 and whose counter value is zero. If two or more slaves with state 1 are having counter values equal to zero, master will give equal priority to those slaves. Go to Step1 and the procedure follows. Here, note that if all the counters are at state 1, they will be reset to zero.

Master polls the class1 slave if all the "class2 slave-master" pairs are in 0-0 state, i.e., master polls the class1 sources only if all the class2 sources are unbacklogged. It polls the class1 slaves following the round-robin manner. Here also, the polling can be explicit polling or implicit polling. Thus, a "Priority Queueing at master with Priority Polling" model can give end-to-end QoS gaurantees by effectively using the bandwidth.

This model can be used to provide end-to-end QoS gaurantees in a Bluetooth scatternet. We consider simple scatternet structure shown in Fig. 3.13.

In the figure, two piconets are connected by a node called relay node. Let source A sending class2 packets to the node B. The relay node forwards the packets from one piconet to another piconet. By using the above model, master M1 polls a backlogged class2 source A in an exhaustive manner. When the source is unbacklogged, it stops polling and forwards all the packets in the priority queue to the relay node. Once

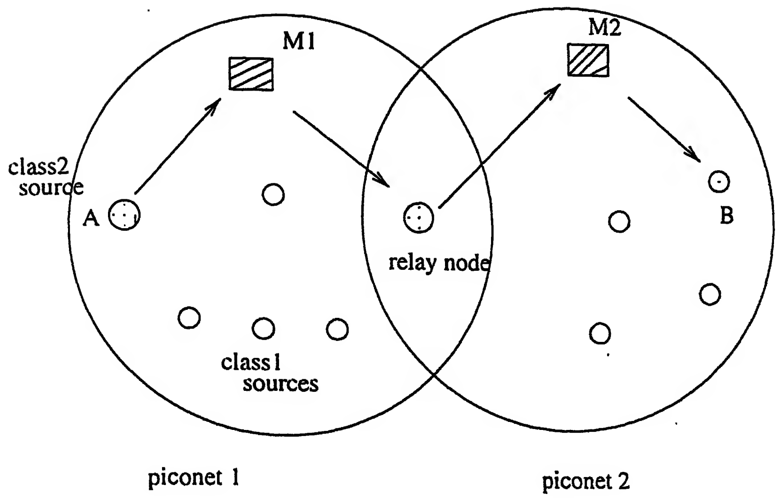


Figure 3.13: A simple scatternet stucture

the priority queue becomes empty, it again polls the class2 source if it is backlogged and forwards the packets to relay node. This process continues until master finds an unbacklogged class2 source when it comes back to poll the slaves after forwarding the packets to relay node. Then, the master polls the other class1 slaves present in the piconet if the class2 source is unbacklogged. Similarly, M2 also implements the same procedure if we assume the relay node as a class2 source for piconet 2. The relay node enters into HOLD or PARK mode in the current piconet when it is visiting the adjacent piconet. Here relay node forwards packets from piconet 1 to piconet 2, when the master M1 is polling the class2 source or other class1 sources. Master will not forward the packets when the relay node is in PARK mode. Instead, it polls the class1 slaves until the relay node is active in the piconet. Similarly the master M2 will not poll the relay node if it is in PARK mode. It serves other class1 slaves present in the piconet until relay is active in the piconet.

Chapter 4

Simulation Results

We now present simulation results for various scheduling policies derived in chapter 3. The two important parameters that have been considered are minimizing end-to-end packet delivery delay and providing consistent data throughput. Discrete Event Simulations have been performed for the various MAC scheduling policies.

4.1 Piconet with 'PQ at master' model:

In the first part of our work, we consider a single piconet of the following type: Piconet consists of seven slaves and a master. Master can be a Fixed Access point or a Base station. Slaves are receiving data packets coming from the master. The actual traffic sources may be located at any point in the Internet. For the purpose of our simulation, we have equivalently assume that the sources are present at the master and account for the wired part of network. Slaves are also generating packets destined for master. Here, the traffic can be either class1 or class2.

We simulate a piconet consisting of seven active slaves and a master. For each slave, there is a corresponding queue at the mater. The data arrival process at the master and slave queues is assumed to be Poisson. The service time of a data packet depends on the packet length. A packet size is chosen uniformly from 1, 3 and 5 slot lengths with equal probability. We keep the buffer sizes sufficiently large to hold the packets. Discrete event simulations were run for 10,000 TDD slots. The TDD slot

length in Bluetooth is equal to 625 μsecs.

The arrival rates (packets/sec) at the slaves are

$$\lambda_{s0} = \lambda_{s1} = \lambda_{s2} = \lambda_{s3} = \lambda_{s4} = \lambda_{s5} = \lambda_{s6} = \lambda_u/7$$

The overall packet arrival rate in the uplink is defined as λ_u. For the overall packet arrival rate λ_u (packets/sec), the offered load in the uplink is given by $\rho_u = \lambda_u \overline{X}$.

Downlink packets destined for each slave arrive at the master with arrival rates (packets/sec)

$$\lambda_{ms0} = \lambda_{ms1} = \lambda_{ms2} = \lambda_{ms3} = \lambda_{ms4} = \lambda_{ms5} = \lambda_{ms6} = \lambda_d/7$$

The overall packet arrival rate in the downlink is defined as λ_d. For the overall packet arrival rate λ_d (packets/sec), the offered load in the downlink is given by $\rho_d = \lambda_d \overline{X}$. For this piconet, the delay performance with RR and 'PQ at master' model is given in the following sections.

4.1.1 Simulation Results with Round-Robin (RR) scheduling:

Fig. 4.1 and 4.2 shows the mean delays for the packets in the uplink and downlink, respectively with the Round-Robin (RR) scheduling. The simulation results have been compared with the expected delays obtained by analytical modeling. We define the mean delay of packets to be the delay incurred from the time it is enqueued in the baseband buffer to when it is received.

To obtain the analytical results, Eqs. 6, 7 are used for the uplink and Eqs. 13, 14 for the downlink. Table 4.1 will show the values of various parameters appear in the equations for both uplink and downlink.

From Figs. 4.1, 4.2, both analytical and simulation results show that offered load should be less than 0.5 for mean delays to be bounded. The reason for this can be explained as follows: In the uplink, each packet transmission is preceded by a polling interval of average length $\overline{V}$, thereby effectively increasing the average transmission time from $\overline{X}$ to $\overline{X} + \overline{V}$. Similarly, in the downlink, each packet transmission is preceded by an idle period or receiving periods from the slaves of average length $\overline{I}$, thereby effectively increasing the average transmission time from $\overline{X}$ to $\overline{X} + \overline{I}$.

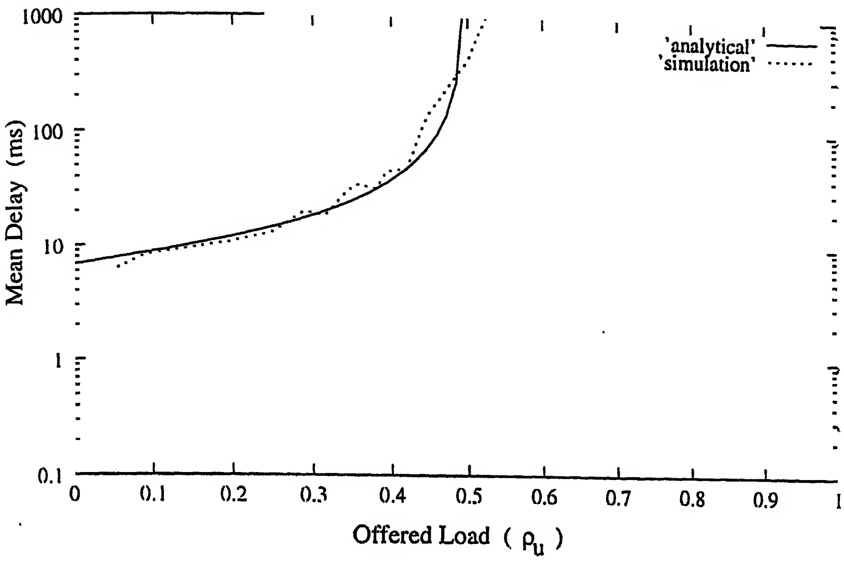


Figure 4.1: Mean Delay Vs Offered Load in the uplink with RR scheduling

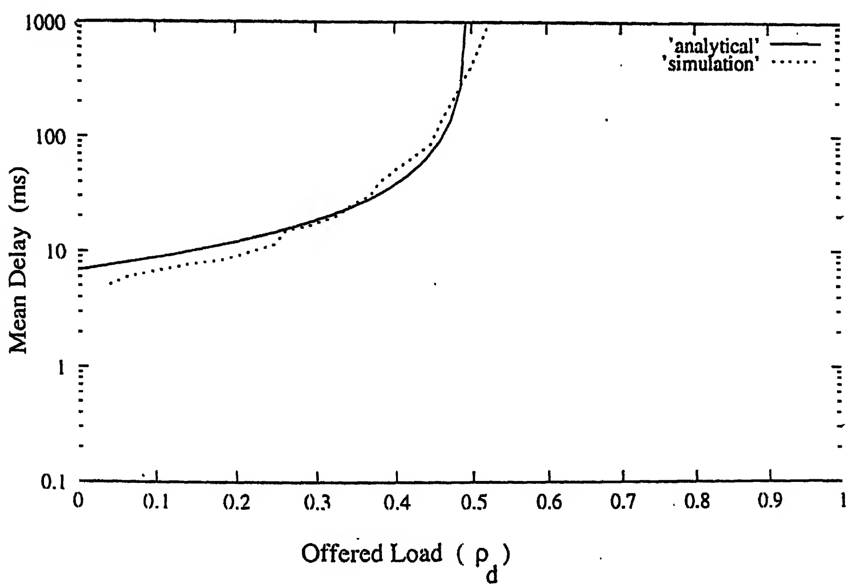


Figure 4.2: Mean Delay Vs Offered Load in the downlink with RR scheduling

Table 4.1: Values of parameters used to obtain analytical results for mean delays in both uplink and downlink with RR scheduling

Paramater	Va ue
m	7
$\overline{X}$	0.001875
$\overline{X^2}$	4.55729e-06
$\overline{V}(uplink)$	0.001875
$\overline{V^2}(uplink)$	4.55729e-06
$\overline{I}(downlink)$	0.001875
$\overline{I^2}(downlink)$	4.55729e-06
$\sigma_V^2(uplink)$	1.041666e-06
$\sigma_I^2(downlink)$	1.041666e-06

4.1.2 Simulation Results with 'Priority Queueing (PQ) at master' model:

In this section, we show the mean delays with 'Priority Queueing (PQ) at master' model. With this model, master maitains priority queue for class2 packets and the scheduling of packets is done as it has been explained in section 3.3. Each class2 packet could be destined for any of the slaves. We assume uniform distribution of the destination. We have calculated the mean delays for class2 packets in the downlink using the Eqs. 20, 21. Table 4.2 gives the values of parameters of the equations. In Fig. 4.3, simulation results have been compared with the analytical results.

Table 4.2: Values of the parameters used to obtain analytical results for mean delays for class2 packets with the 'PQ at master' model

Paramater	Value
$\overline{X}$	0.001875
$\overline{X_e^2}$	16.145833e-06
$\overline{V_e}$	0.00375
$\overline{V_e^2}$	16.145833e-06

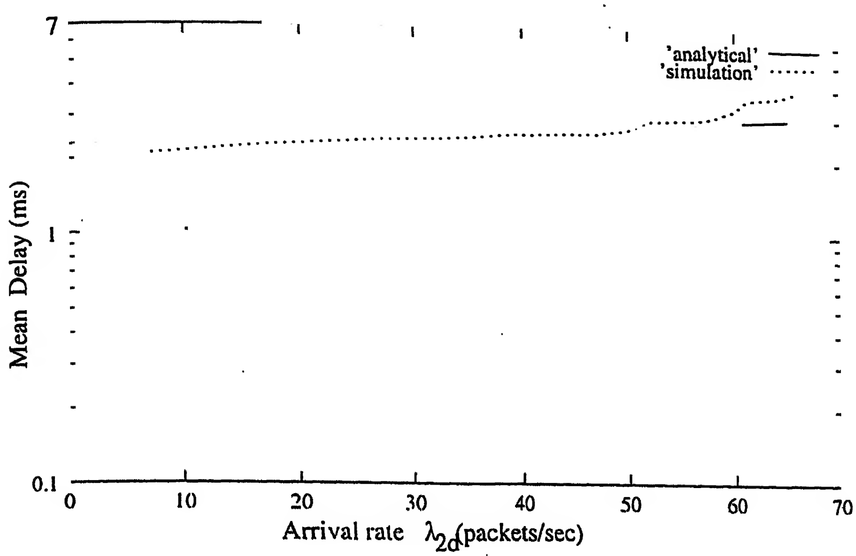


Figure 4.3: Mean Delay Vs Arrival Rate for the class2 packets in a downlink with a 'PQ at master' model

Fig. 4.4 shows the improvement in mean delays for class2 packets in the downlink with the 'PQ at master model' over the delays with the conventional Round-Robin scheduling. Here, note that half of the offered load ρ_d is from class1.

Fig. 4.5 shows the mean delays for class2 packets in downlink with PQ at master model when the half of the offered load, ρ_d, is from class1. Figure shows the mean delays for class1 also. From the figure, class1's delay goes to infinity at around offered load $\rho_d = 0.68$. This means that class1 traffic is rarely transmitted for ρ_d greater than 0.68. The class2's delay is bounded even at the offered loads equal to one.

Fig. 4.6 shows the throughput values for both class1 and class2 in downlink with 'PQ at master' model. Here, throughput is defined as the rate (packets/sec) at which packets undergone service. From the figure, it can be observed that beginning at load $\rho_d = 0.7$, the throughput for class1 starts to decrease while that of class2 continues to increase. This is reasonable since half of the offered load is from class2 and the priority is given to class2 packet transmission.

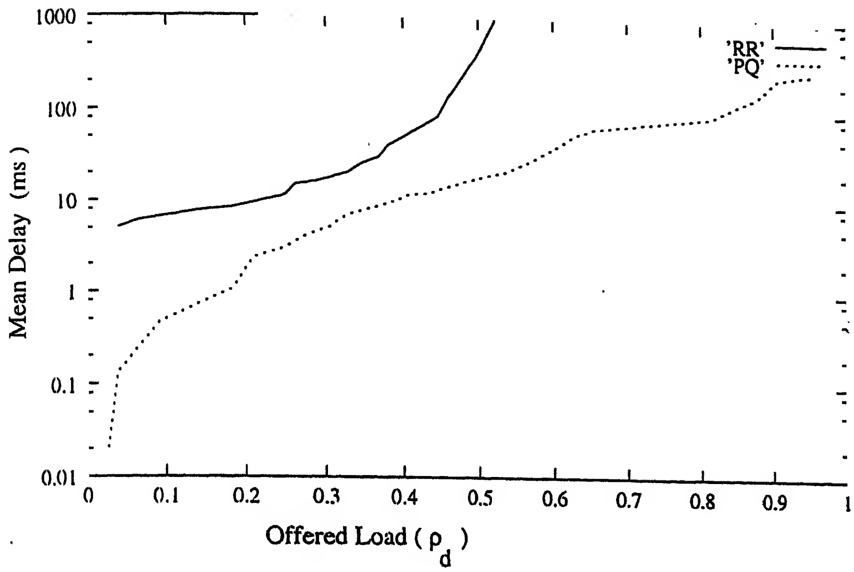


Figure 4.4: Mean delay Vs Offered Load for class2 packets in the downlink with RR and 'PQ at master' model

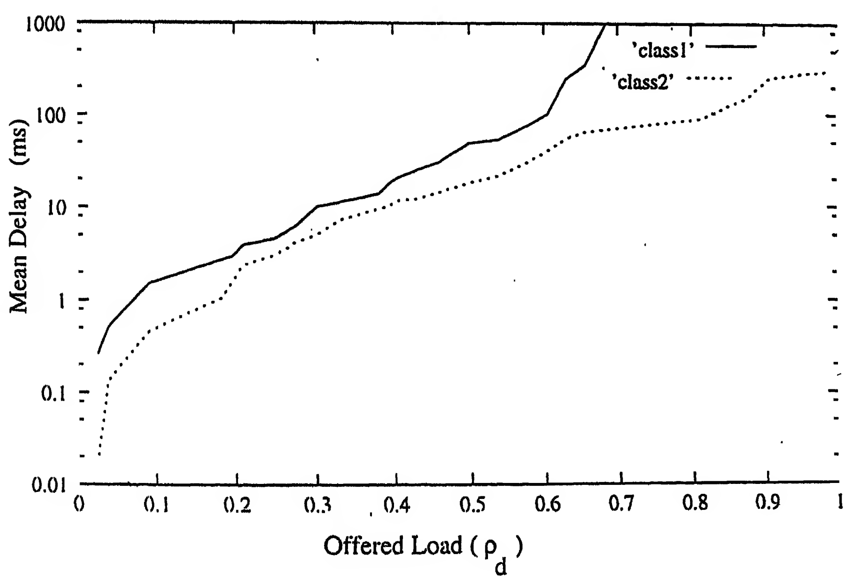


Figure 4.5: Mean Delay Vs Offered Load for both class1 and class2 in downlink with 'PQ at master' model

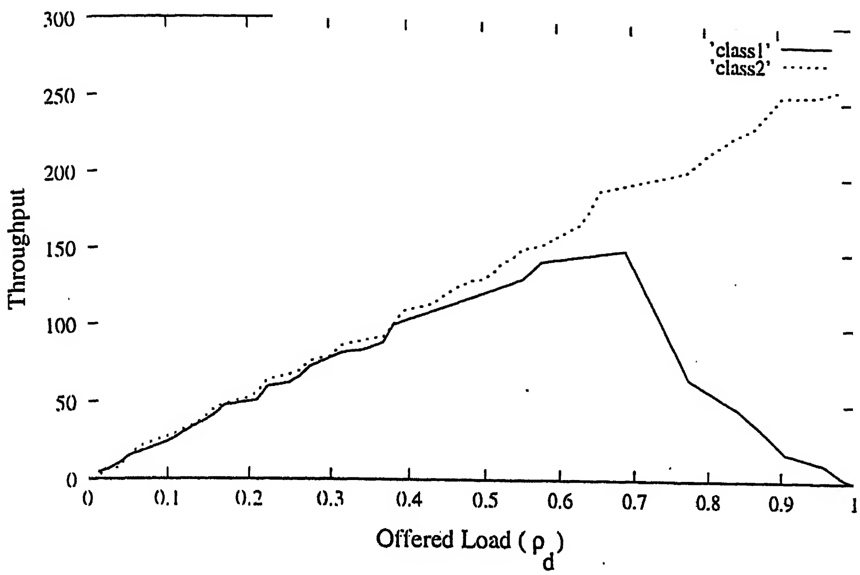


Figure 4.6: Throughput Vs Offered Load for both class1 and class2 in downlink with 'PQ at master' model

4.2 Piconet with 'PQ at master with PP' model:

In the second part, we simulate a piconet consisting of seven slaves and master. Slaves are communicating with other slaves through the master. Here, master acts as a router which receives the packets from the slaves and fowards them to the destinations. We have devided the slaves into two groups. First group consists of slaves generating class2 packets and the second group consists of slaves generating class1 packets. A 'Priority Queueing (PQ) at master with Priority Polling (PP)' model has been used in these piconets. The scheduling of packets is done as it is explained in section 3.4. Here, master maintains a prioirty output queue for class2 packets. Also, for each slave, there is a corresponding output class1 queue at master. The data arrival process at the slaves is assumed to be Poisson. A packet size is chosen uniformly from 1, 3 and 5 slot with equal probability. Discrete Event Simulations are run for 10,000 TDD slots. Simulations have been carried out by varying the number of class2 sources.

The arrival rates (packets/sec) at the slaves are

$$\lambda_{s0} = \lambda_{s1} = \lambda_{s2} = \lambda_{s3} = \lambda_{s4} = \lambda_{s5} = \lambda_{s6} = \lambda/7$$

The overall packet arrival rate in a piconet is defined as λ. For the overall packet arrival rate λ (packets/sec), the offered load in the piconet is given by $\rho = \lambda\overline{X}$.

4.2.1 Simulation Results with 'PQ at master with PP' model

First, we consider a single class2 source. Fig. 4.7 shows the mean delay for class2 packets in the uplink and Fig. 4.8 shows the mean delays for class2 in the downlink. Analytical results have been compared with the simulation results. Eqs. 32, 33 are used to obtain the analytical results. Table 4.3 shows the values of various parametrs that appear in the equations.

Table 4.3: Values of the parameters used to obtain analytical results for mean delays for class2 packets with the 'PQ at master with PP' model

Paramater	Value
$\overline{X}$	0.001875
$\overline{X_e}$	0.00375
$\overline{X_e^2}$	16.145833e-06
$\overline{V_e}$	0.00375
$\overline{V_e^2}$	16.145833e-06

Fig. 4.9 shows the improvement in delay for class2 with this model over RR scheduling for uplink and Fig. 4.10 shows the mean end-to-end delay for both class1 and class2 in a piconet consisting of one class2 source and six class1 sources.

Next, we increase the number of class2 sources in a piconet to three and observed the delay performance with the model. So, in this piconet there are three class2 sources and four class1 sources. Scheduling of packets is done as it has been explained in section 3.4. Fig. 4.11 shows the delay performance for both class1 and class2. It can observed from the figure that the delay for class2 packets is increased as the offered load from class2 sources is more in this case. Fig. 4.12 shows the throughput of class1 and class2 in this piconet. From the figure throughput for class2 continues to increase as the offered load in a piconet increases, whereas the throughput for class1 is decreasing starting at around ρ (Offered Load) = 0.55.

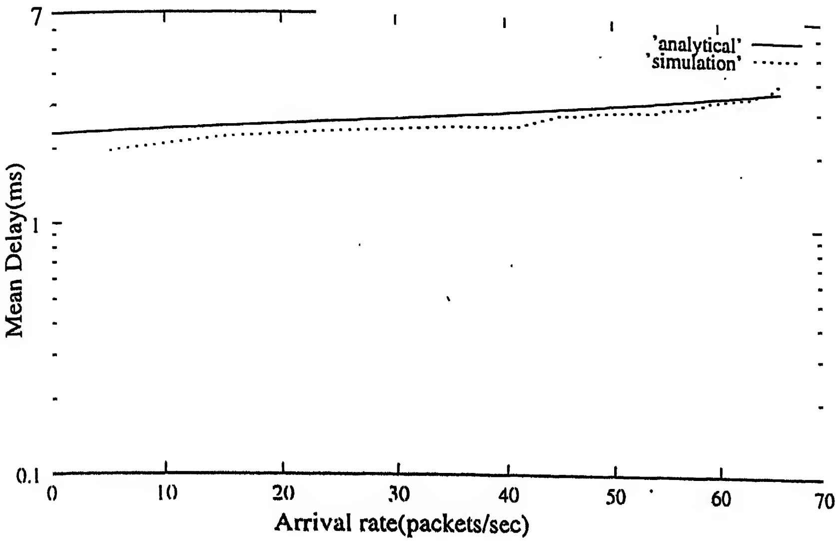


Figure 4.7: Mean Delay Vs Arrival rate (λ_2) for class2 in the uplink with 'PQ at master with PP' model

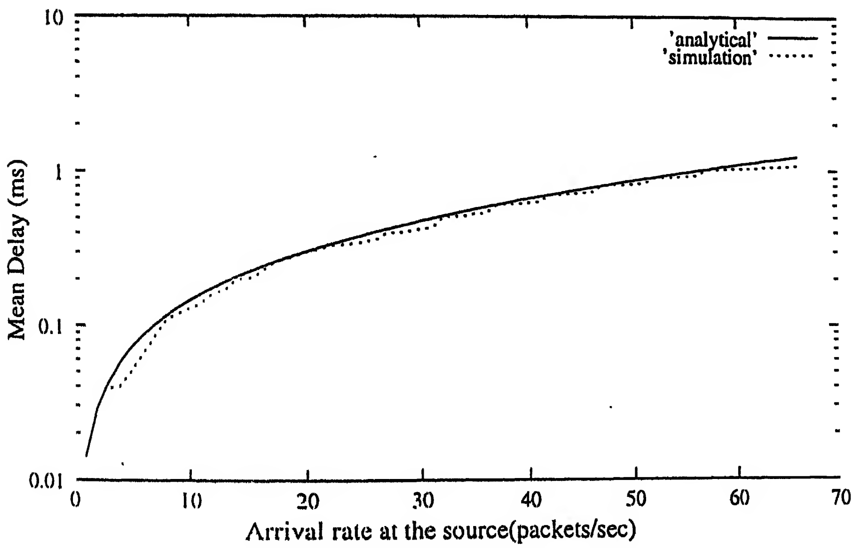


Figure 4.8: Mean Delay Vs Arrival rate (λ_2) for class2 in the downlink with 'PQ at master with PP' model

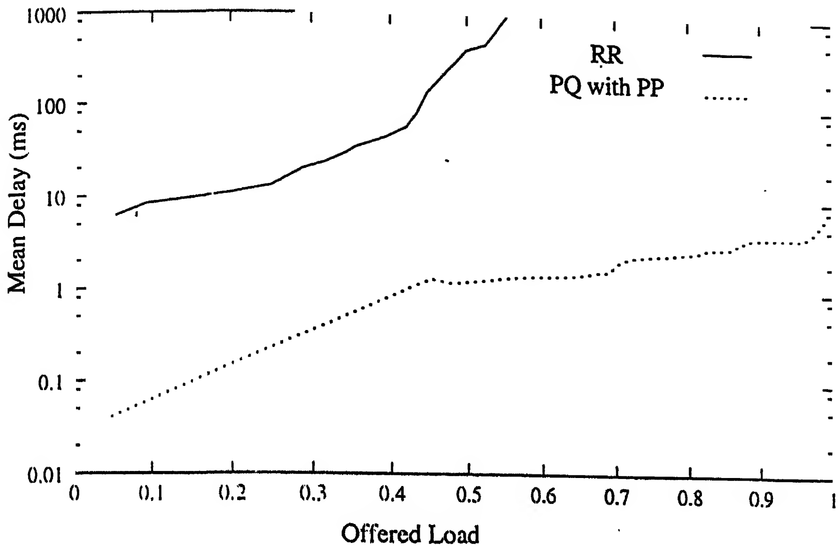


Figure 4.9: Mean Delay Vs Offered Load for class2 in the uplink with RR and 'PQ at master with PP' model

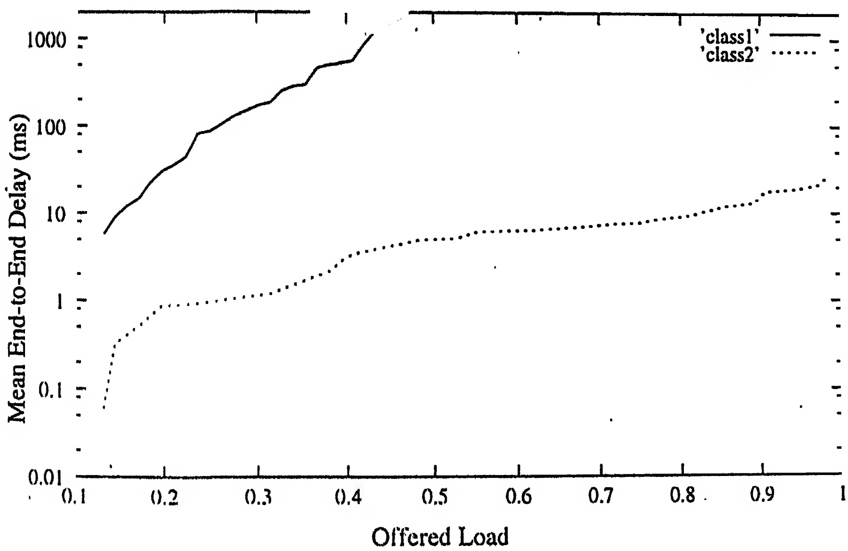


Figure 4.10: Mean End-to-End delay Vs Offered Load for both class1 and class2 with 'PQ at master with PP' model

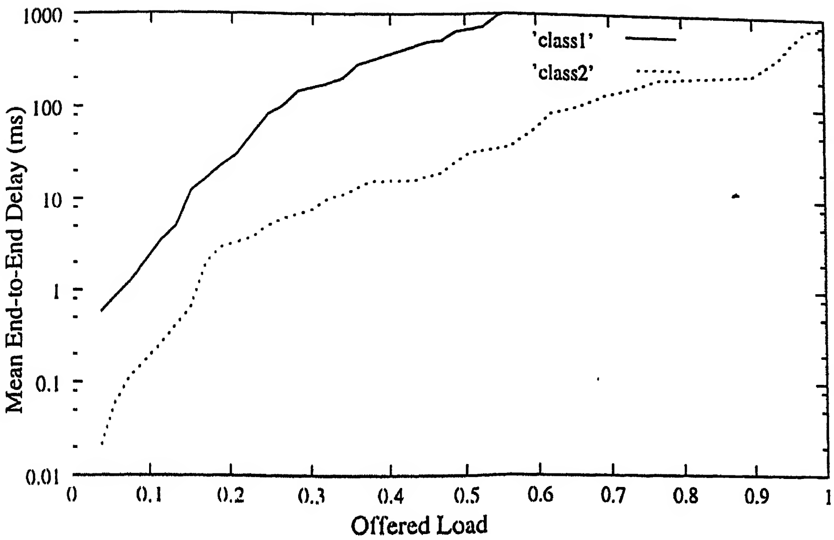


Figure 4.11: Mean End-to-End Delay Vs Offered Load for both class2 and class1 packets in a piconet consisting of three class2 sources and four class1 sources with 'PQ at master with PP' model

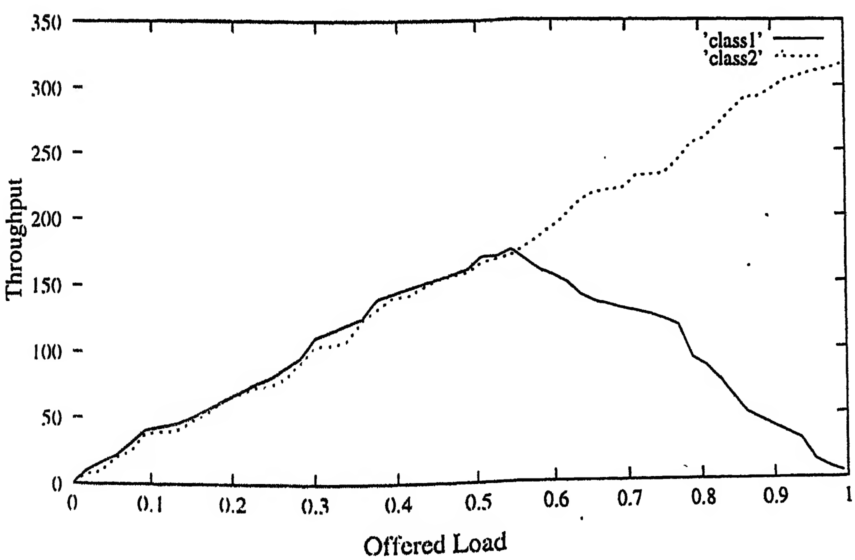


Figure 4.12: Throughput Vs Offered Load for both class1 and class2 in piconet consisting of three class2 sources and four class1 sources with 'PQ at master with PP' model

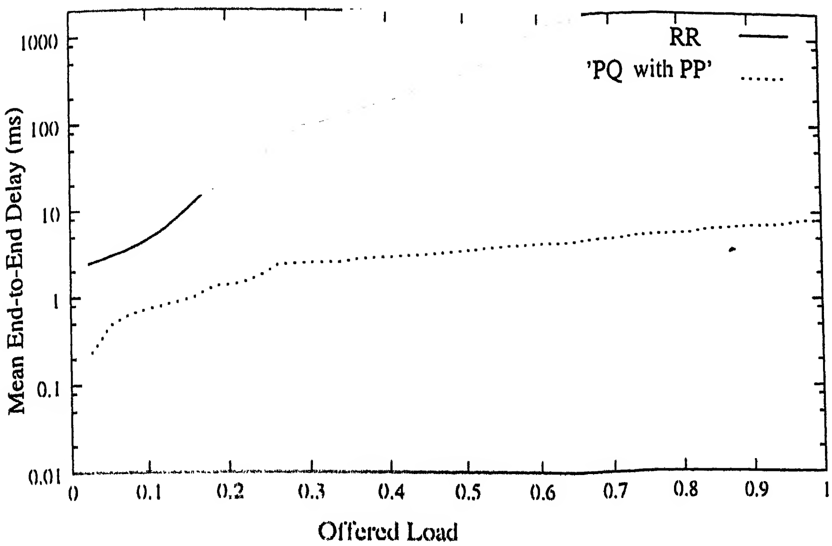


Figure 4.13: Mean End-to-End Delay Vs Offered Load for class2 in a scatternet with RR and 'PQ at master with PP' model

From the above results, a 'PQ at master with PP polling' model gives end-to-end QoS guarantees by minimizing end-to-end packet delivery and providing consistent data throughput for the delay sensitive traffic like class2. The performance of class1 is observed to be deteriorating at higher offered loads from class2.

The 'PQ at master with PP model' can also be used to provide QoS gaurantees in a Bluetooth scatternet. We have simulated a simple scatternet structure consisting of two piconets connected by a relay node. We consider the case of single class2 source sending packets to the destination present in the adjucent piconet. The relay node forwards the packets from the source piconet to destination piconet. Fig 4.13 shows the Mean End-to-End delay for the class2 with RR scheduling and with 'PQ at master with PP model'. From the figure, the delay performance for class2 with 'PQ at master with PP' model outperforms the delay performance with RR scheduling.

Chapter 5

Conclusion

A non-real-time communication service in Bluetooth was considered. Depending on its distinct characteristics and QoS requirements, the non-real-time traffic can be divided into two classes: a) class1: delay-tolerant traffic like paging and email; and b) class2: delay-sensitive traffic like FTP and remote log-in. The motivation behind this work was to support different classes of service in a Bluetooth system. The objective was to minimize the end-to-end delay and providing consistent data throughput for a delay-sensitive traffic like class2 traffic.

A breif overview of Bluetooth was presented in Chapter 2. In Chapter 3 we have derived the expressions for mean delay performances with various scheduling policies by analytical modeling. We have considered two types of piconets. One corresponds to the master connected to wired infrastructure and acting as a Base station. In this piconet slaves are receiving data through the master. A 'Priority Queueing (PQ) at master' model has been proposed to support the classes of service in this type of piconet. An M/G/1 queue with vacations model was used to derive the mean delay expression for the packets in a class2 priority queue at the master. The second type of piconet was an adhoc system with no wired infrastructure, where master acting as a router receives the packets from the slaves and forwards to the destinations. In this sytem slaves are communicating with other slaves through the master. A 'Priority Queueing (PQ) at master with Priority Polling (PP)' model has been proposed to support classes of service in this type of piconet. An M/G/1

queue with vacations, where the first vacation period has a different distribution than that of succeeding vacation periods, was used to derive the mean end-to-end delay expression for class 2 packets.

In Chapter 4, we showed the simulation results for the delay and throughput performance with the proposed models. Simulation results were compared with the analytical results. We observe that a simple MAC scheduling algorithm such as Round-Robin is not suitable for Bluetooth as it is unable to minimize the delay for class2 kind of traffic. A 'PQ at master' model minimizes the delay for class2 traffic in the downlink. With this model, throughput for class2 is continuously increasing but the throughput for class1 starts decreasing at around offered load $\rho_d = 0.7$ (see Fig. 4.5), where the half of offered load is from class2. A 'PQ at master with PP' model gives end-to-end QoS gaurantees by minimizing end-to-end delay packet delivery delay and providing consistent data throughput for class2 traffic. The throughput for class1 starts decreasing at around $\rho(loadinapiconet) = 0.55$, see Fig 4.12, where there are three class2 sources out of seven sources. Further, it is shown that a 'PQ at master with PP' model can also be used to provide end-to-end QoS gaurantees in a Bluetooth scatternet.

5.1 Future Work:

In this work, we have analyzed the delay and throughput performance by assuming the arrival process of packets as a Poisson process and uniform distribution of packet size. So the proposed models can be evaluated in a more real environment. In the future work, the performance of the proposed models can be studied by incorporating wireless link failures. Further study is required to incorporate low power modes (*sniff, hold, park*) into MAC scheduling and to explore the effect of varying number of slaves in a piconet.

Bibliography

[1] Bluetooth Specification Version 1.0A, http://www.bluetooth.com/.

[2] J.C. Haartsen, The Bluetooth Radio System, IEEE*Personal Communications*, pp. 28-36, Feb2000.

[3] C.E. Perkins and E.M. Royer, Ad Hoc On Demand Distance Vector (AODV) Routing, Internet Draft, March 2000. http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-05.txt/.

[4] Z.j. Haas and M.R. Pearlman, The Performance of a New Routing Protocol for the Reconfigurable Wireless Networks, Proc. ICC '98., pages 156-160.

[5] J. Broch, D.B. Johnson and D.A. Maltz, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, Internet Draft, October 1999. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt

[6] P. Bhagwat and A. segall, A Routing Vector Method(RVM) for Routing in Bluetooth Scatternets, *in The Sixth IEEE International Workshop on Mobile Multimedia Communicatins* - MoMuC'99, pp. 375-379, 1999.

[7] J. Zyren, Reliability of IEEE 802.11 Hi Rate DSSS WLANs in a High Density Bluetooth Environment, White paper, Harris Semiconductors, June 1999.

[8] J.C. Haartsen and S. Zurbes, Bluetooth Voice and Data Performance in 802.11 DSWLAN Environment, White paper, Erricsson, May 1999.

[9] D.Suvak, Comparing the Benefits of IrDA and Bluetooth, Wireless Systems Design, May 2000.

[10] M. Kalia, D. Bansal and R. Shorey, MAC Scheduling and SAR Policies for Bluetooth: A Master Driven TDD Pico-Cellular Wireless System, *in The Sixth IEEE International Workshop on Mobile Multimedia Communications* - Mo Muc'99, pp. 375-379, San Diego, California, November 1999.

[11] Sughyun Choi and Kang G.shin, A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communication Services, IEEE/ACM *Transactions on Networking*, Vol. 8,No. 1,pp. 44-59, Feb 2000.

[12] D.Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1994.

[13] R. Ait Yaiz, G. Heijenk, Polling in Bluetooth a Simplified Best Effort Case, *in the Seventh Annual CTIT Workshop*, pp. 91-96, The Netherlands, Feb. 2001.

[14] L. Kleinrock, *Queueing Systems*, Vol1. New York: Wiley, 1975.

Appendix A

Suppose that we observe the system from time t = 0 to the indefinite future and we record the values of various quantities of interest as time progress. In particular, let N(t) be the Number of packets in the system at time t, $\alpha(t)$ be the number of packets which are arrived in the interval [0,t] and T_i be the time spent by the i^{th} arriving packet. Number of packets in the system observed up to time t (N_t) is given by

$$N_t = \frac{1}{t}\int_0^t N(\tau)d\tau \tag{A.1}$$

Naturally N_t changes with the time t, but in many systems of interest, N_t tends to a staedy-state N as t increases, that is

$$N = \lim_{t\to\infty} N_t \tag{A.2}$$

In this case, we call N the steady-state time average of $N(\tau)$. The time average arrival rate over the interval [0,t], λ_t is given by

$$\lambda_t = \frac{\alpha(t)}{t} \tag{A.3}$$

The steady-state arrival rate is defined as $\lambda = \lim_{t\to\infty} \lambda_t$. The time average of packet delay up to time t is similarly defined as

$$T = \frac{\sum_{i=0}^{\alpha(t)} T_i}{\alpha(t)} \tag{A.4}$$

that is, the avearge time spent in the system per packet up to time t. The steady-state time average packet delay is defined as

$$T = \lim_{t \to \infty} T_t \tag{A.5}$$

It turns out that the quantities N, λ, and T above are related by a simple formula that makes it possible to determine one given the other. This result, known as Little's theorem, has the form

$$N = \lambda T \tag{A.6}$$

Appendix B

Expression for the mean queueing delay of a packet in the uplink given in section 3.1.1 contains the term Y, the mean duration of polling intervals. The following section presents the derivation of the term Y.

B.1 Mean duration of polling intervals in the uplink with RR scheduling

The system shown in Figure 3.1 can be modeled as a limited-1, partially gated system given in [12]. The expression for Y in limited-1, partially gated system can be obtained from the expression for Y in exhaustive system [12]. In an exhaustive system, each slave sends all the packets waiting in it's output queue in the data intervals. So we first calculate Y for an exhaustive system. Denote

$$\beta_{lj} = E\{Y_i/packetarrivesinslavel'spollingordatainterval$$

$$andbelongstoslave(l+j)modm\}$$

We have,

$$\beta_{lj} = 0, j = 0$$

$$\beta_{lj} = \overline{V}_{(l+1)modm} + \ldots + \overline{V}_{(l+j)modm}, j > 0$$

Since packet i belongs to any slave with equal probability 1/m, we have

$$E\{Y_i/packetarrivesinslavel'spollingordatainterval\}$$

$$= \frac{1}{m}\sum_{j=1}^{m-1}\beta_{lj} = \sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V}_{(l+j)modm} \tag{B.1}$$

In steady-state, a packet will arrive during slave l's data interval with probability ρ_u/m, and during slave l's polling interval with probability

$$\frac{(1-\rho_u)\overline{V}_l}{\sum_{k=0}^{m-1}\overline{V}_k}$$

Using this fact in Equ. B.1, we obatin the following equation for $Y = \lim_{i\to\infty} E\{Y_i\}$:

$$Y = \frac{\rho_u}{m}\sum_{j=1}^{m-1}\frac{m-j}{m}\sum_{l=0}^{m-1}\overline{V}_l + \frac{1-\rho_u}{\sum_{k=0}^{m-1}\overline{V}_k}\sum_{l=0}^{m-1}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V}_l\overline{V}_{(l+j)modm} \tag{B.2}$$

The last sum above can be written

$$\sum_{l=0}^{m-1}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V}_l\overline{V}_{(l+j)modm} = \frac{1}{2}\left[\left(\sum_{l=0}^{m-1}\overline{V}_l\right)^2 - \sum_{l=0}^{m-1}\overline{V}_l^2\right]$$

Using this expression and denoting

$$\overline{V} = \frac{1}{m}\sum_{l=0}^{m-1}\overline{V}_l$$

As the polling interval averaged over all slaves, we can write Eq. B.2 as

$$Y = \frac{(m-\rho_u)\overline{V}}{2} - \frac{(1-\rho_u)\sum_{l=0}^{m-1}\overline{V}_l^2}{2m\overline{V}} \tag{B.3}$$

We now consider a limited-1, partially gated system whereby, in each slave's data interval, only the first packet of slave waiting in queue is transmitted. We argue as follows. A packet arriving during slave l's data or polling interval will belong to any one of the slaves with equal probability 1/m. Therefore, in steady-state, the expected number of packets waiting in the queue of the slave that owns the arriving packet, averaged over all slaves, is $\lim_{i\to\infty} E\{N_i\}/m = \lambda_u \frac{W_{qu}}{m}$. Each of these packets causes an extra cycle of polling intervals $m\overline{V}$, so Y is increased by an amount $\lambda_u W_{qu}\overline{V}$.

So the mean duration of polling intervals Y, is given by

$$Y = \frac{(m-\rho_u)\overline{V}}{2} - \frac{(1-\rho_u)\sum_{l=0}^{m-1}\overline{V}_l^2}{2m\overline{V}} + \lambda_u W_{qu}\overline{V} \tag{B.4}$$

Similarly, expression for the mean queueing delay of a packet in the downlink given in section 3.1.2 contains the term Y, the mean duration of idle intervals. The following section presents the derivation of the term Y.

B.2 Mean duration of idle intervals in the downlink with RR scheduling

The system shown in Figure 3.4 can be modeled as a limited-1, partially gated system given in [12]. The expression for Y in limited-1, partially gated system can be obtained from the expression for Y in exhaustive system [12]. In an exhaustive system, master sends all the packets waiting in it's output queue in the data intervals. So we first calculate Y for an exhaustive system. Denote

$$\beta_{lj} = E\{Y_i/packetarrivesinmasterqueuel'sidleor$$

$$datainterval and belongstomasterqueue(l + j)modm\}$$

We have,

$$\beta_{lj} = 0, j = 0$$

$$\beta_{lj} = \overline{V}_{(l+1)modm} + \ldots + \overline{V}_{(l+j)modm}, j > 0$$

Since packet i belongs to any slave with equal probability 1/m, we have

$$E\{Y_i/packetarrivesinmasterqueuel'sidleordatainterval\}$$

$$= \frac{1}{m}\sum_{j=1}^{m-1}\beta_{lj} = \sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V}_{(l+j)modm} \qquad (B.5)$$

In steady-state, a packet will arrive during master queue l's data interval with probability ρ_u/m, and during master queue l's idle interval with probability

$$\frac{(1-\rho_d)\overline{V}_l}{\sum_{k=0}^{m-1}\overline{V}_k}$$

Using this fact in Equ. B.5, we obatin the following equation for $Y = \lim_{i\to\infty} E\{Y_i\}$:

$$Y = \frac{\rho_d}{m} \sum_{j=1}^{m-1} \frac{m-j}{m} \sum_{l=0}^{m-1} \overline{V}_l + \frac{1-\rho_u}{\sum_{k=0}^{m-1} \overline{V}_k} \sum_{l=0}^{m-1} \sum_{j=1}^{m-1} \frac{m-j}{m} \overline{V}_l \overline{V}_{(l+j) mod m} \tag{B.6}$$

The last sum above can be written

$$\sum_{l=0}^{m-1} \sum_{j=1}^{m-1} \frac{m-j}{m} \overline{V}_l \overline{V}_{(l+j) mod m} = \frac{1}{2} \left[\left(\sum_{l=0}^{m-1} \overline{V}_l \right)^2 - \sum_{l=0}^{m-1} \overline{V}_l^2 \right]$$

Using this expression and denoting

$$\overline{V} = \frac{1}{m} \sum_{l=0}^{m-1} \overline{V}_l$$

As the polling interval averaged over all slaves, we can write Eq. B.6 as

$$Y = \frac{(m-\rho_d)\overline{V}}{2} - \frac{(1-\rho_d) \sum_{l=0}^{m-1} \overline{V}_l^2}{2m\overline{V}} \tag{B.7}$$

We now consider a limited-1, partially gated system whereby, in each master queue's data interval, only the first packet waiting in queue is transmitted. We argue as follows. A packet arriving during master queue l's data or polling interval will belong to any one of the queues of master with equal probability $1/m$. Therefore, in steady-state, the expected number of packets waiting in the queue of the master that owns the arriving packet, averaged over all queues, is $\lim_{i \to \infty} E\{N_i\}/m = \lambda_u \frac{W_{qu}}{m}$. Each of these packets causes an extra cycle of idle intervals $m\overline{V}$, so Y is increased by an amount $\lambda_u W_{qu} \overline{V}$.

So the mean duration of idle intervals Y, is given by

$$Y = \frac{(m-\rho_d)\overline{V}}{2} - \frac{(1-\rho_d) \sum_{l=0}^{m-1} \overline{V}_l^2}{2m\overline{V}} + \lambda_d W_{qd} \overline{V} \tag{B.8}$$

Appendix C

In section 3.4, the mean queueing delay of a class2 packet at class2 slave (see Eq. 3.32) contains the terms $\overline{B}$, $\overline{B^2}$. The following section presents the derivation of these terms.

C.1 The Busy Period in a M/G/1 queue:

In section 3.4, the queue of a class2 slave is modeled as a M/G/1 queue with vacations, where the distribution of first vacation period is exceptional than the distributions of succeeding vacation periods. A Busy Period starts with the arrival of a packet to an otherwise empty queue and continues until the queue becomes empty once again. We let B be the random variable denoting the length of such a busy period with pdf $f_B(t)$ whose Laplace Transform (L.T) is $F_B(s)$.

We can assume the service discipline to be LCFS (Last Come First Serve) without changing the distribution of the busy period (even though the queue is/may be FCFS (First Come First Serve) in nature). This is because of that once the packets are in the queue, we can permute the order in which they are served without changing the distribution of the busy period. We therefore consider the busy period for LCFS queue in the following and will subsequently use the result obtained for FCFS queue. This is used to derive its pdf $f_B(t)$ and its L.T. $F_B(s)$. Consider a busy period which starts with the arrival of packet A_1. Let,

- X_{e1} be the service time for A_1.

- n arrivals $(A_2, \ldots\ldots\ldots\ldots, A_{n+1})$ arrive during the service time X_{e1}, in that sequence.

- B is the length of busy period started by A_1.

Consider the n arrivals $(A_2, \ldots\ldots, A_{n+1})$ during the service time X_{e1} of A_1. Since the service is LCFS in nature, each one these will initiate sub-busy periods $B_2, \ldots\ldots, B_{n+1}$ so that we have

$$B = X_{e1} + B_2 + \ldots + B_{n+1} \tag{C.1}$$

Where B_j is independent of B_k and X_{e1}.
The sub-busy periods are i.i.d random variables and the distribution of a sub-busy period will be the same as the distribution of the main busy period.

Then, $E\{e^{-sB}/X_{e1} = x, n = k\} = e^{-sx}[F_B(s)]^k$

Hence,

$$\begin{aligned} E\{e^{-sB}/X_{e1} = x\} &= e^{-sx}\sum_{k=0}^{\infty} e^{-\lambda_2 x}\frac{(\lambda_2 x)^k}{K!}[F_B(s)]^k \\ &= e^{-x(s+\lambda_2-\lambda_2 F_B(s))} \end{aligned} \tag{C.2}$$

Therefore,

$$F_B(s) = E\{e^{-sB}\} = \int_{x=0}^{\infty} e^{-x[s+\lambda_2-\lambda_2 F_B(s)]}b(x)dx$$

or

$$F_B(s) = B(s + \lambda_2 - \lambda_2 F_B(s)) \tag{C.3}$$

Therefore, assuming $\rho_2 = \lambda_2\overline{X_e}$ the first and second moments of Busy Period $\overline{B}$, $\overline{B^2}$ can be found from the above expression which are given by

$$\overline{B} = \frac{\overline{X_e}}{(1-\rho_2)}$$

$$\overline{B^2} = \frac{\overline{X_e^2}}{(1-\rho_2)^3}$$